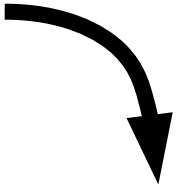
Machine Learning with Scikit-Learn

Andreas Mueller (NYU Center for Data Science, scikit-learn)

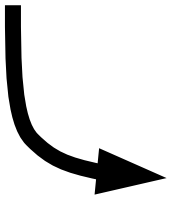
http://bit.ly/sklodsc

Me


universitätbonn
ais




Amazon
Machine
Learning




NYU
Center for
Data Science

Classification
Regression
Clustering
Semi-Supervised Learning
Feature Selection
Feature Extraction
Manifold Learning
Dimensionality Reduction
Kernel Approximation
Hyperparameter Optimization
Evaluation Metrics
Out-of-core learning
……


agramfort
Alexandre Gramfort


AlexanderFabisch
Alexander Fabisch


alextp
Alexandre Passos


amueller
Andreas Mueller


arjoly
Arnaud Joly


bdholt1
Brian Holt


bthirion
bthirion


chrisfilo
Chris Filo Gorgole...


cournape
David Cournapeau


duchesnay
Duchesnay


dwf
David Warde-Farley


fabianp
Fabian Pedregosa


GaelVaroquaux
Gael Varoquaux


glouppe
Gilles Louppe


jakevdp
Jake Vanderplas


jaquesgrobler
Jaques Grobler


jnothman


kastnerkyle
Kyle Kastner


kuantkid
Wei LI


larsmans
Lars


lucidfrontier45
Shiqiao Du


mblondel
Mathieu Blondel


MechCoder
Manoj Kumar


ndawe
Noel Dawe


NelleV
Varoquaux


ogrisel
Olivier Grisel


paolo-losi
Paolo Losi


pprett
Peter Prettenhofer


robertlayton
Robert Layton


ronw
Ron Weiss


satra
Satrajit Ghosh


sklearn-ci

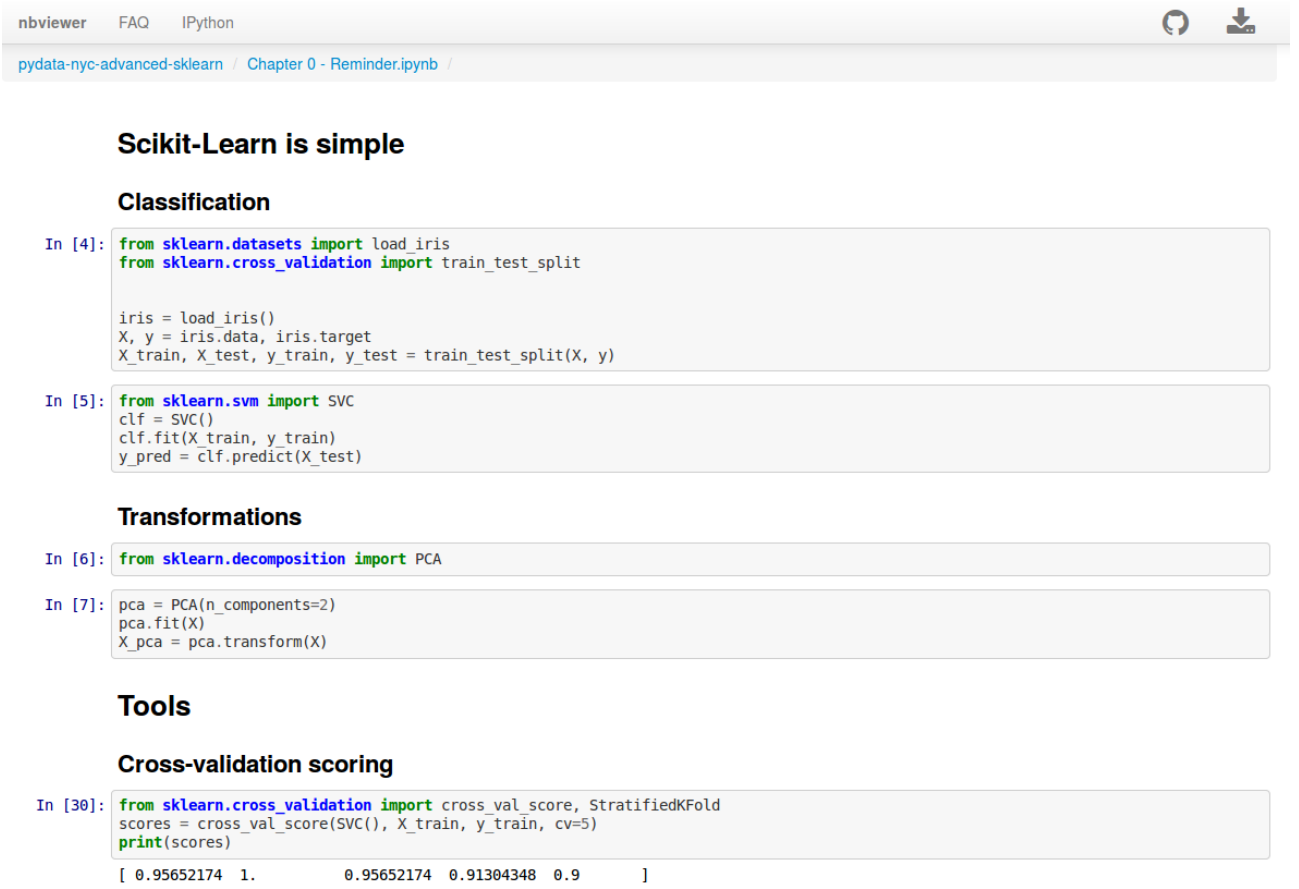

vene
Vlad Niculae


VirgileFritsch
Virgile Fritsch


vmichel
Vincent Michel


yarikoptic
Yaroslav Halchenko

Get the notebooks!



http://bit.ly/sklosdc

scikit learn

Home Installation Documentation Examples

Google™ Custom Search Search

Documentation of scikit-learn 0.16-git

Quick Start

A very short introduction into machine learning problems and how to solve them using scikit-learn. Introduced basic concepts and conventions.

User Guide

The main documentation. This contains an in-depth description of all algorithms and how to apply them.

Other Versions

- scikit-learn 0.15 (stable)
- scikit-learn 0.16 (development)
- scikit-learn 0.14
- scikit-learn 0.13
- scikit-learn 0.12
- Older versions

Tutorials

Useful tutorials for developing a feel for some of scikit-learn's applications in the machine learning field.

API

The exact API of all functions and classes, as given by the docstrings. The API documents expected types and allowed features for all functions, and all parameters available for the algorithms.

Additional Resources

Talks given, slide-sets and other information relevant to scikit-learn.

Contributing

Information on how to contribute. This also contains useful information for advanced users, for example how to build their own estimators.

Flow Chart

A graphical overview of basic areas of machine learning, and guidance which kind of algorithms to use in a given situation.

FAQ

Frequently asked questions about the project and contributing.

http://scikit-learn.org/

Hi Andy,

I just received an email from the first tutorial speaker, presenting right before you, saying he's ill and won't be able to make it.

I know you have already committed yourself to two presentations, but is there anyway you could increase your tutorial time slot, maybe just offer time to try out what you've taught? Otherwise I have to do some kind of modern dance interpretation of Python in data :-)
-Leah

Hi Andreas,

I am very interested in your Machine Learning background. I work for X Recruiting who have been engaged by Z, a worldwide leading supplier of Y. We are expanding the core engineering team and we are looking for really passionate engineers who want to create their own story and help millions of people.

Can we find a time for a call to chat for a few minutes about this?

Thanks

Hi Andy,

I just received an email fr the first tutorial speaker, presenting , saying he's ill and won't be

I know you have a urself to two presentations you could increase you , maybe just offer time to try out what you've taught? Otherwise I have to do some kind of modern dance interpretation of Python in data :-)
-Leah

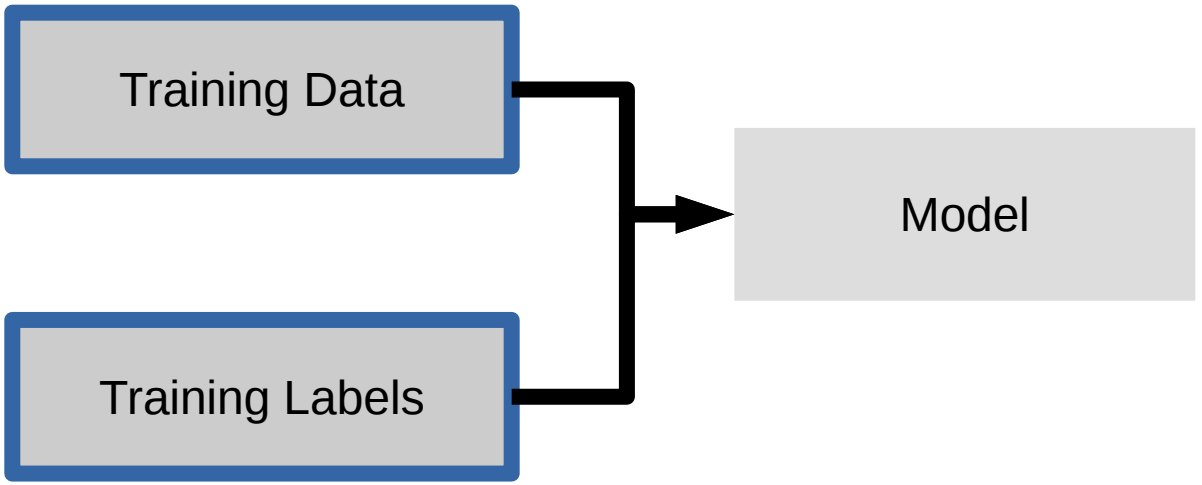
Hi Andreas,

I am very interested in your Machine Learning background. I work for X Recruiting who have been engaged by Z, a worldwide leading supplier of Y. We are expanding the core engineering team and we are looking for really passionate engineers who own story and help millions of

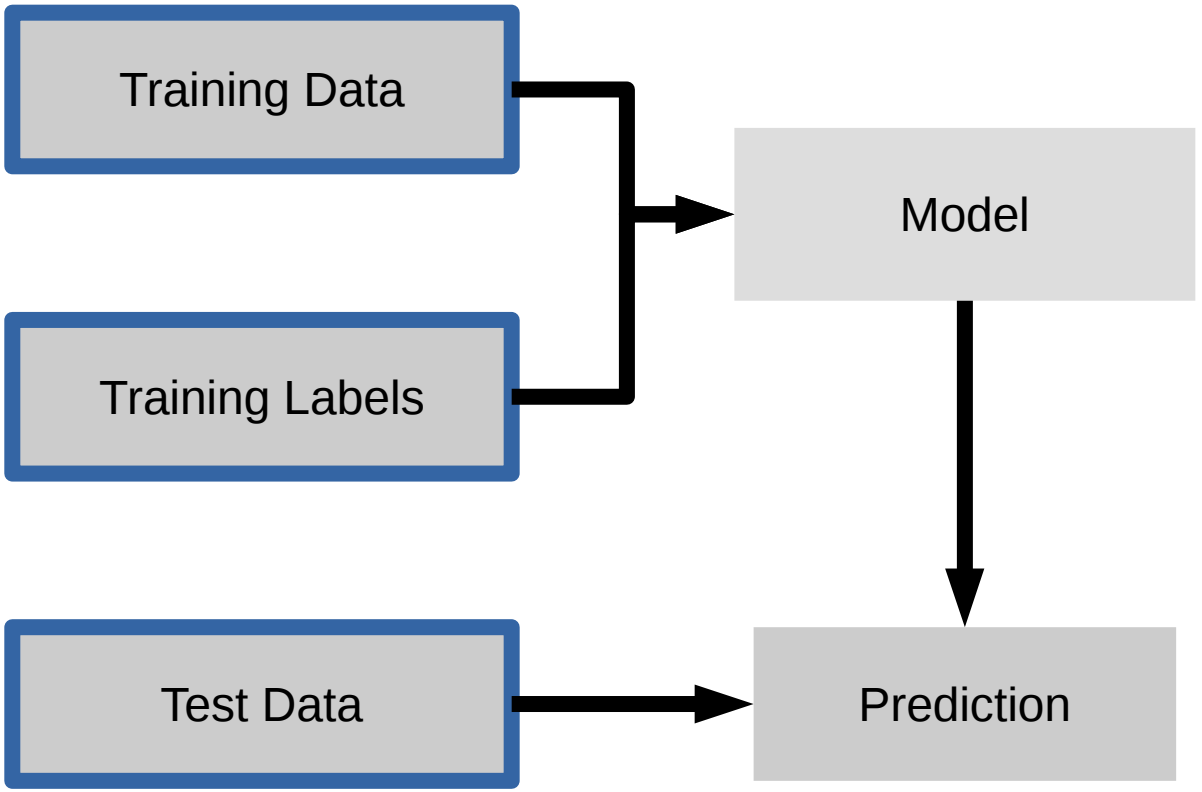
Can we find a ti for a few minutes about t

Thanks

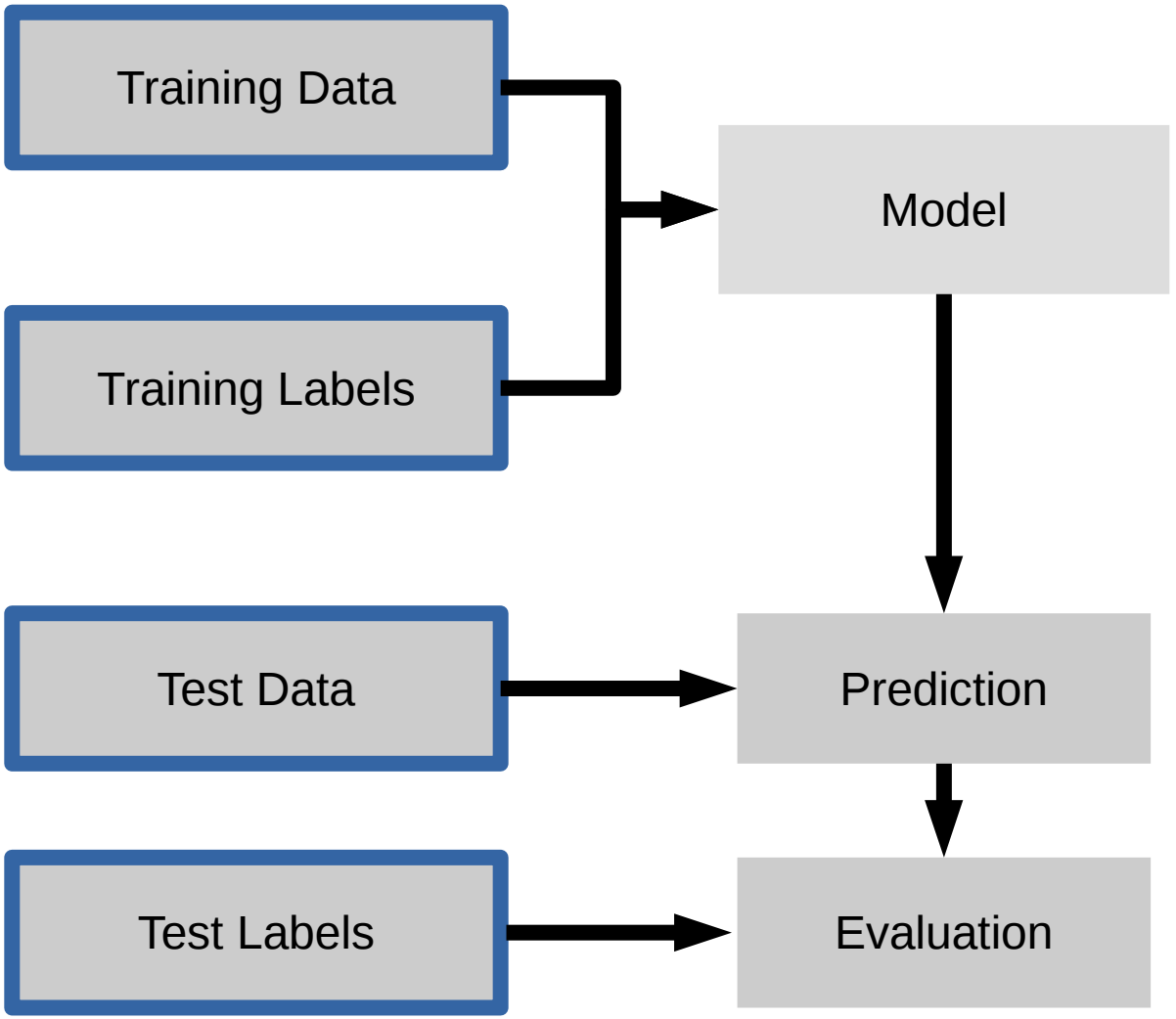
Supervised Machine Learning



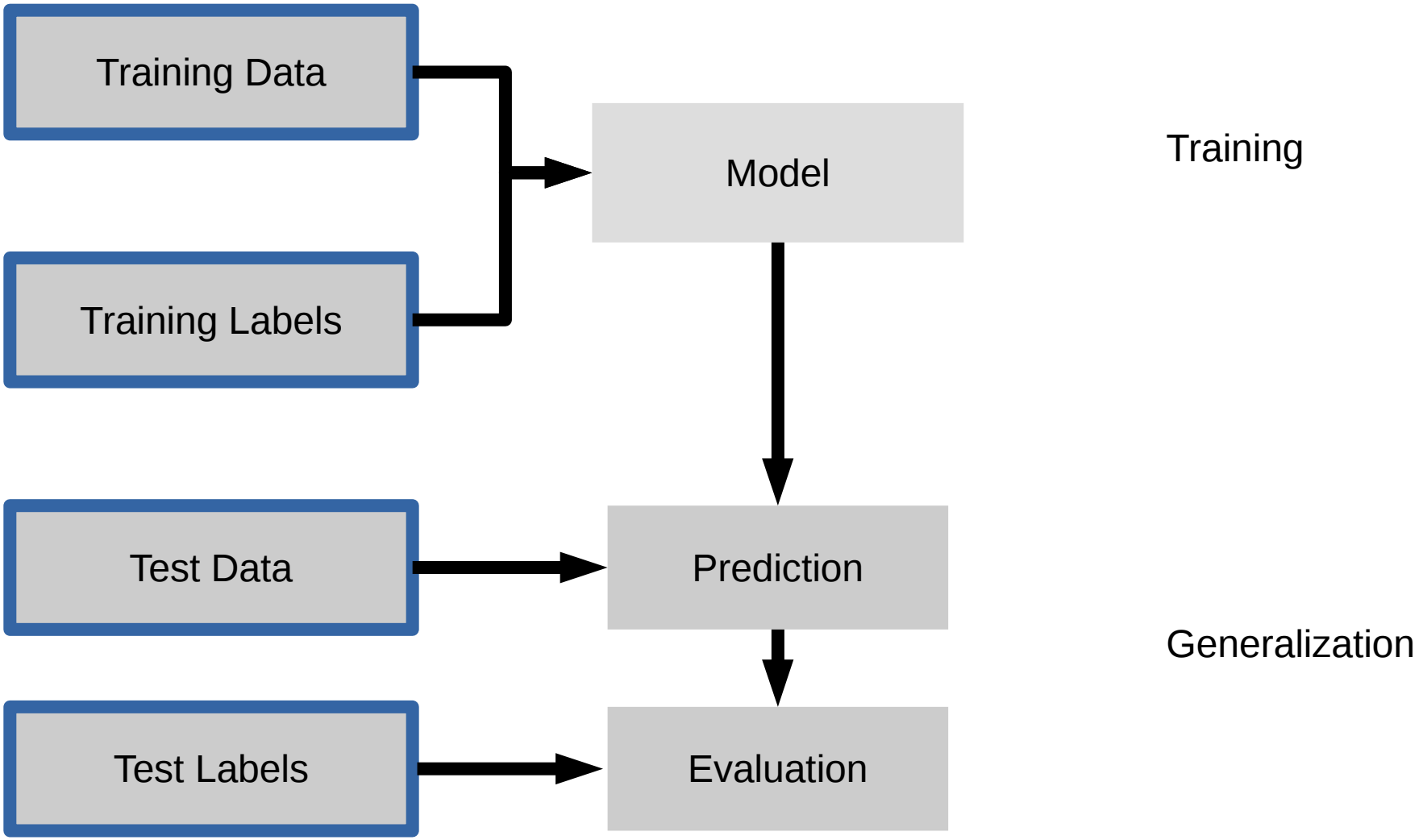
Supervised Machine Learning



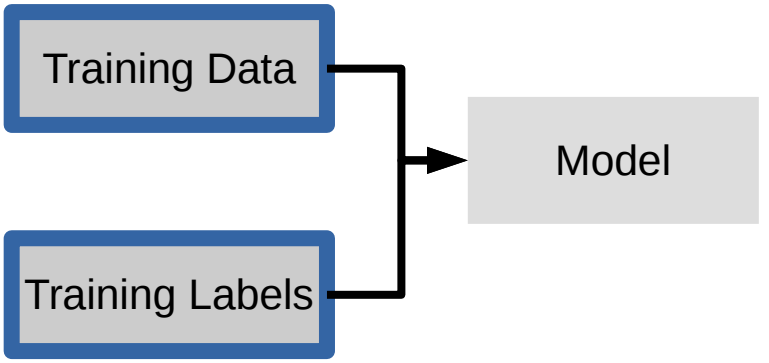
Supervised Machine Learning



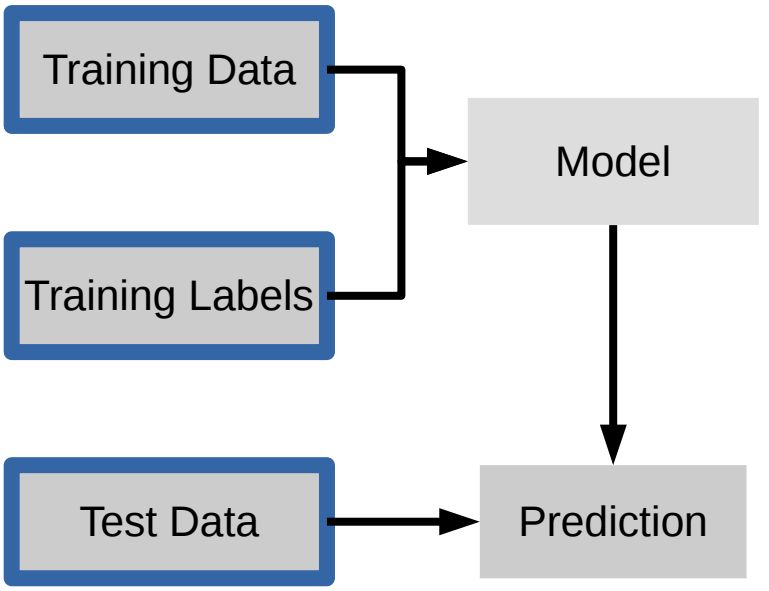
Supervised Machine Learning



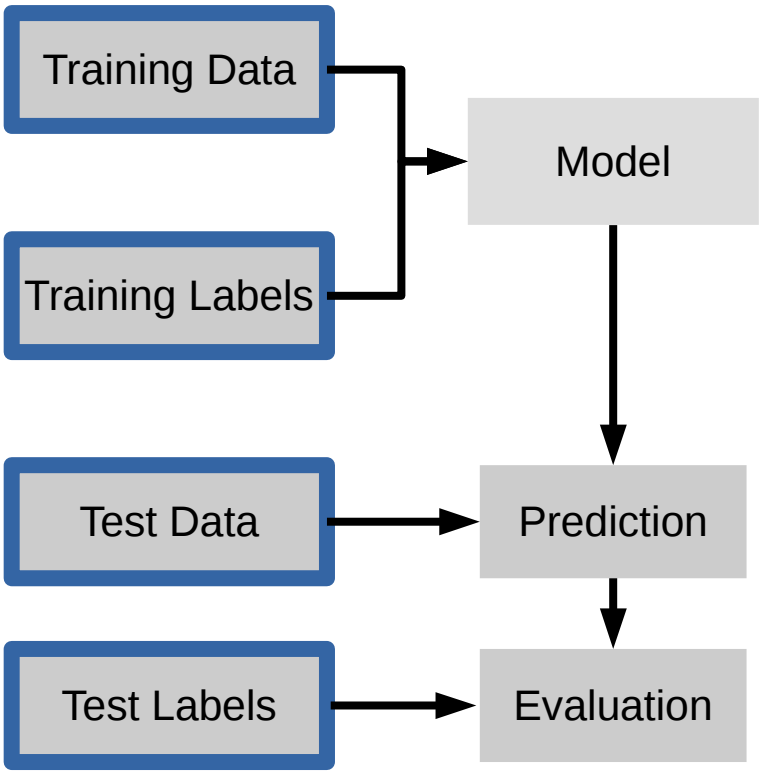
```
clf = RandomForestClassifier()

clf.fit(X_train, y_train)
```



```
clf = RandomForestClassifier()

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
```

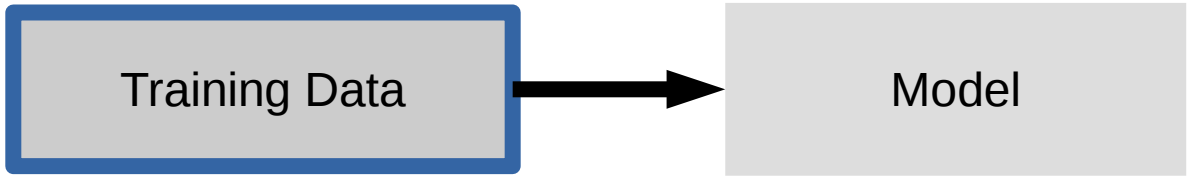


```
clf = RandomForestClassifier()

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

clf.score(X_test, y_test)
```

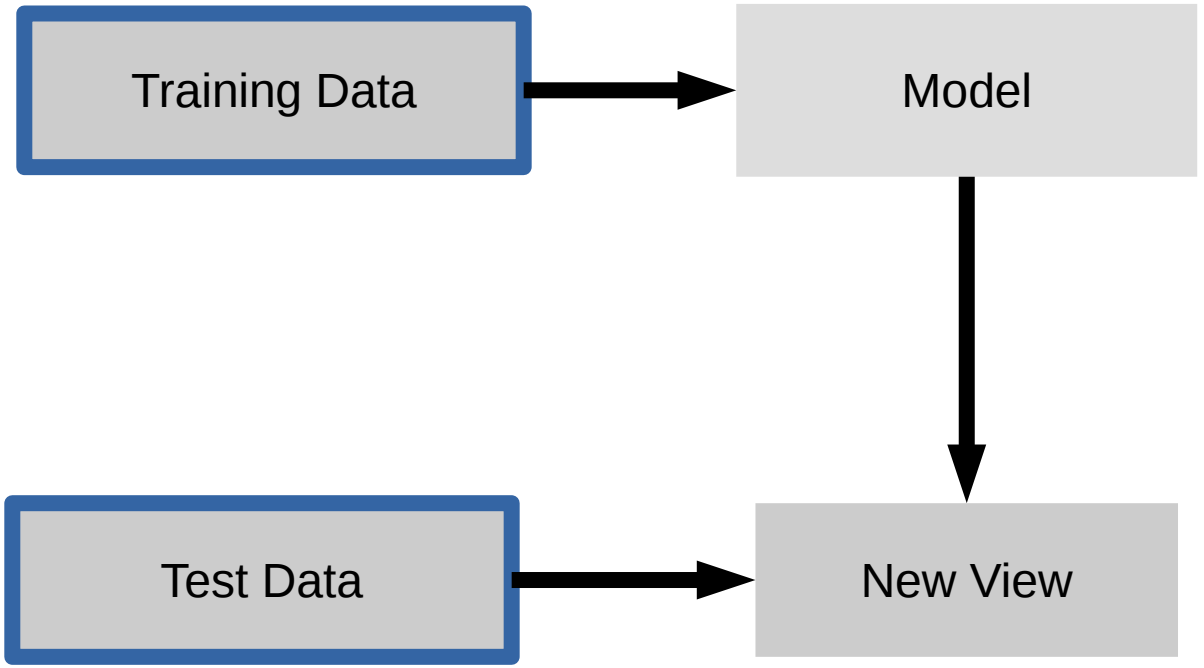


IPython Notebook:
Part 1 - Introduction to Scikit-learn

Unsupervised Machine Learning



Unsupervised Machine Learning

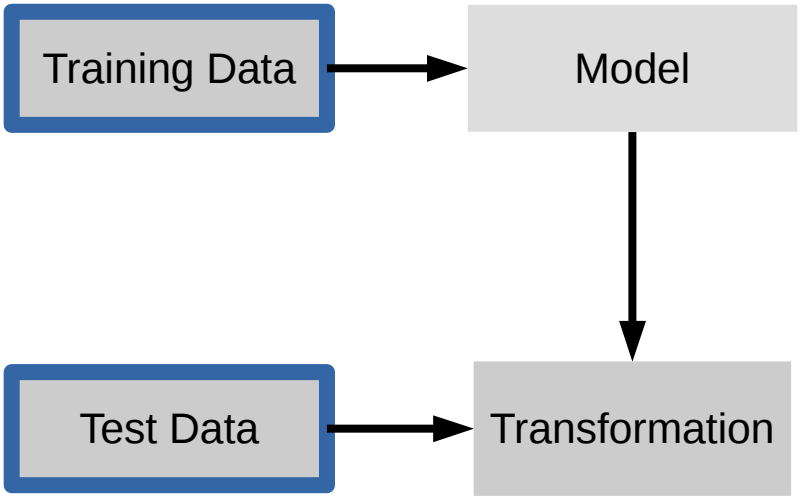


Unsupervised Transformations

```
pca = PCA()
```

```
pca.fit(X_train)
```

```
X_new = pca.transform(X_test)
```



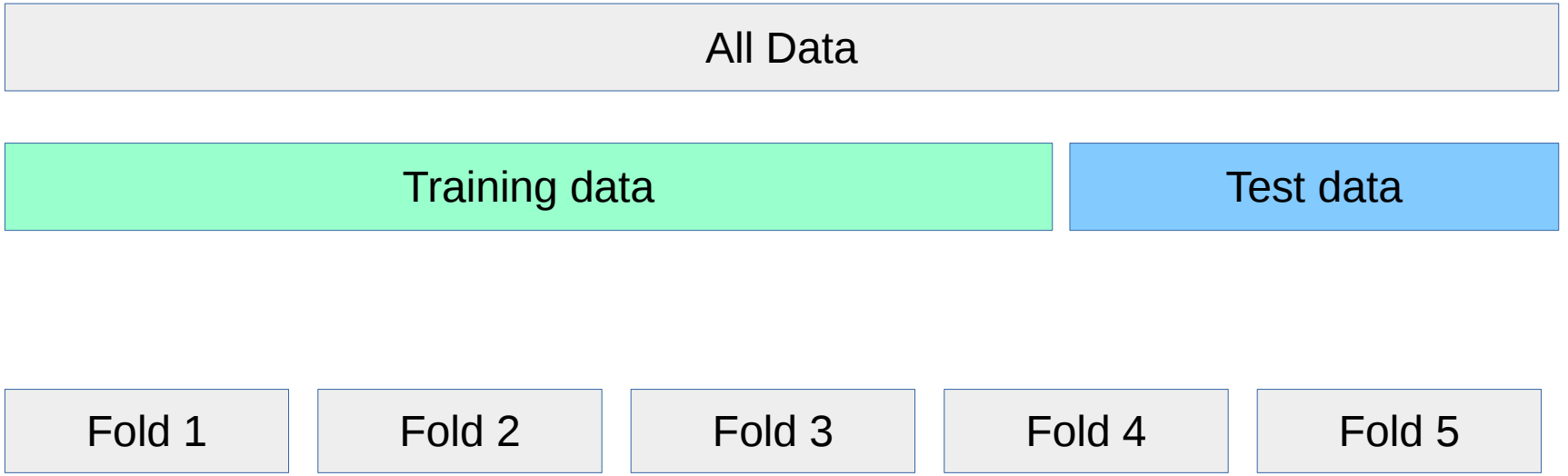
IPython Notebook:
Part 2 – Unsupervised Transformers

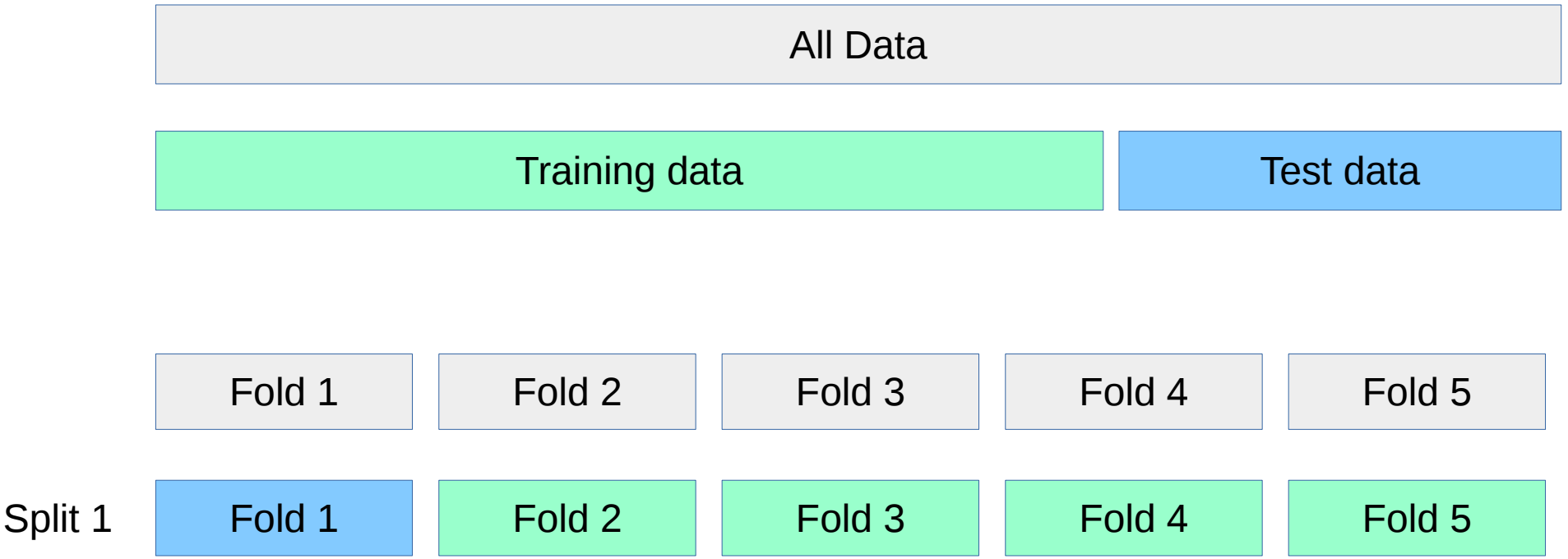
Basic API

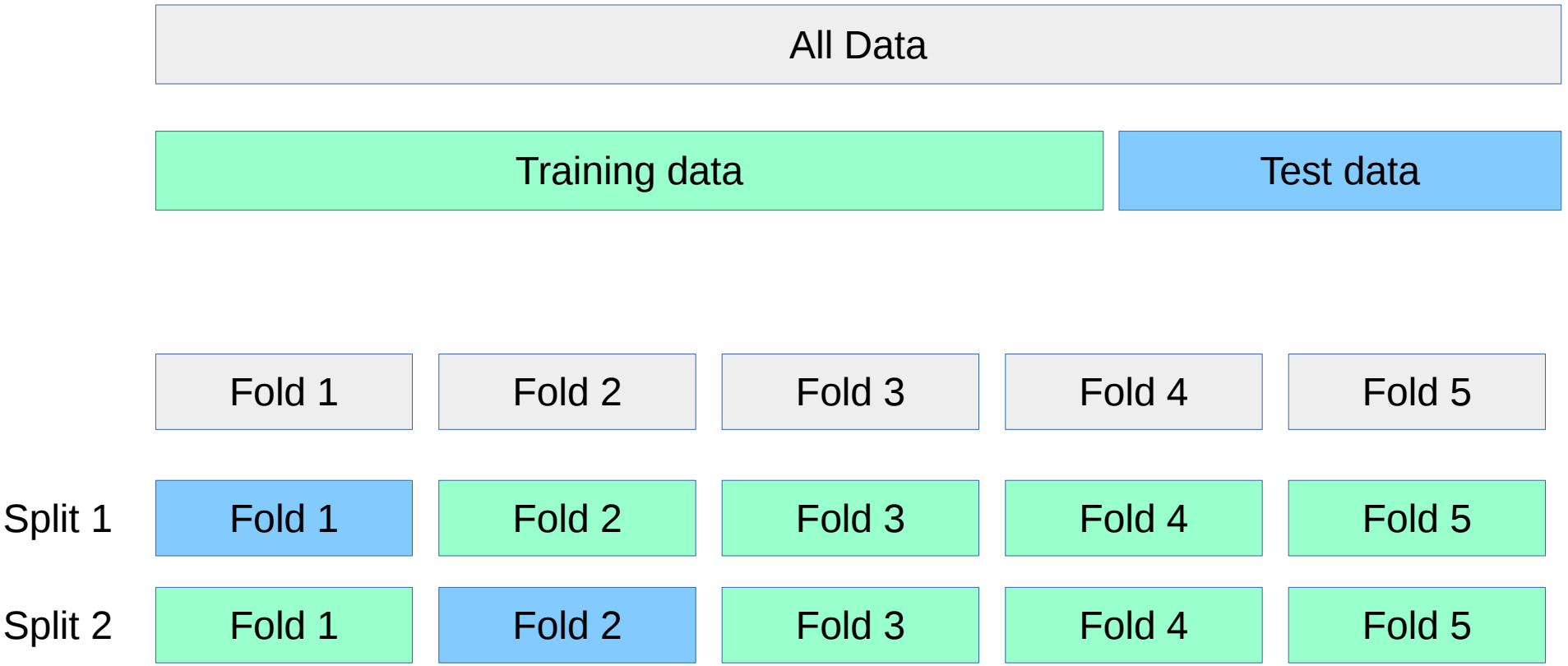
```
estimator.fit(X, [y])
```

`estimator.predict`	`estimator.transform`
Classification	Preprocessing
Regression	Dimensionality reduction
Clustering	Feature selection
	Feature extraction


All Data
Training data
Test data


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5

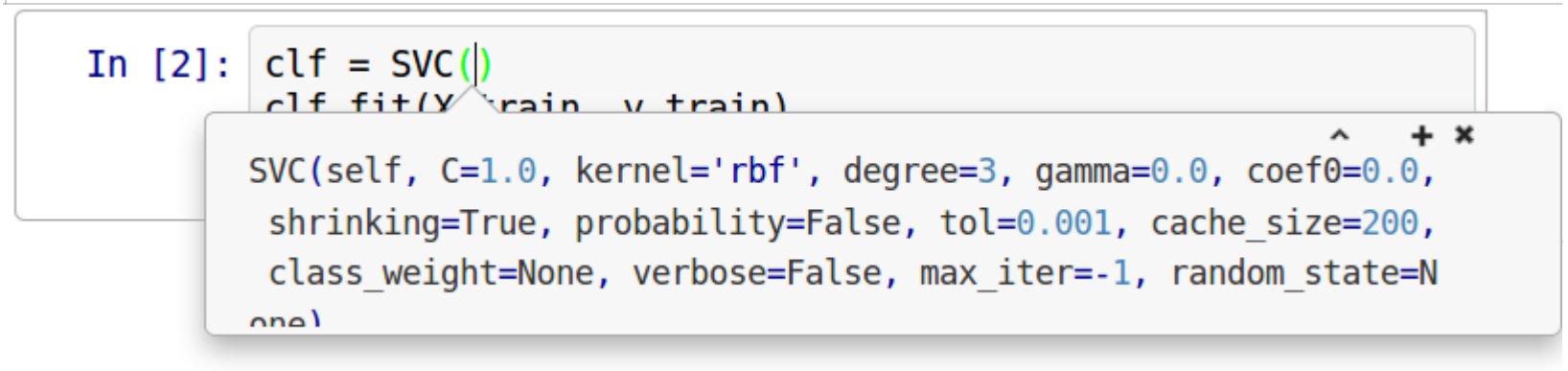

All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 2
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5

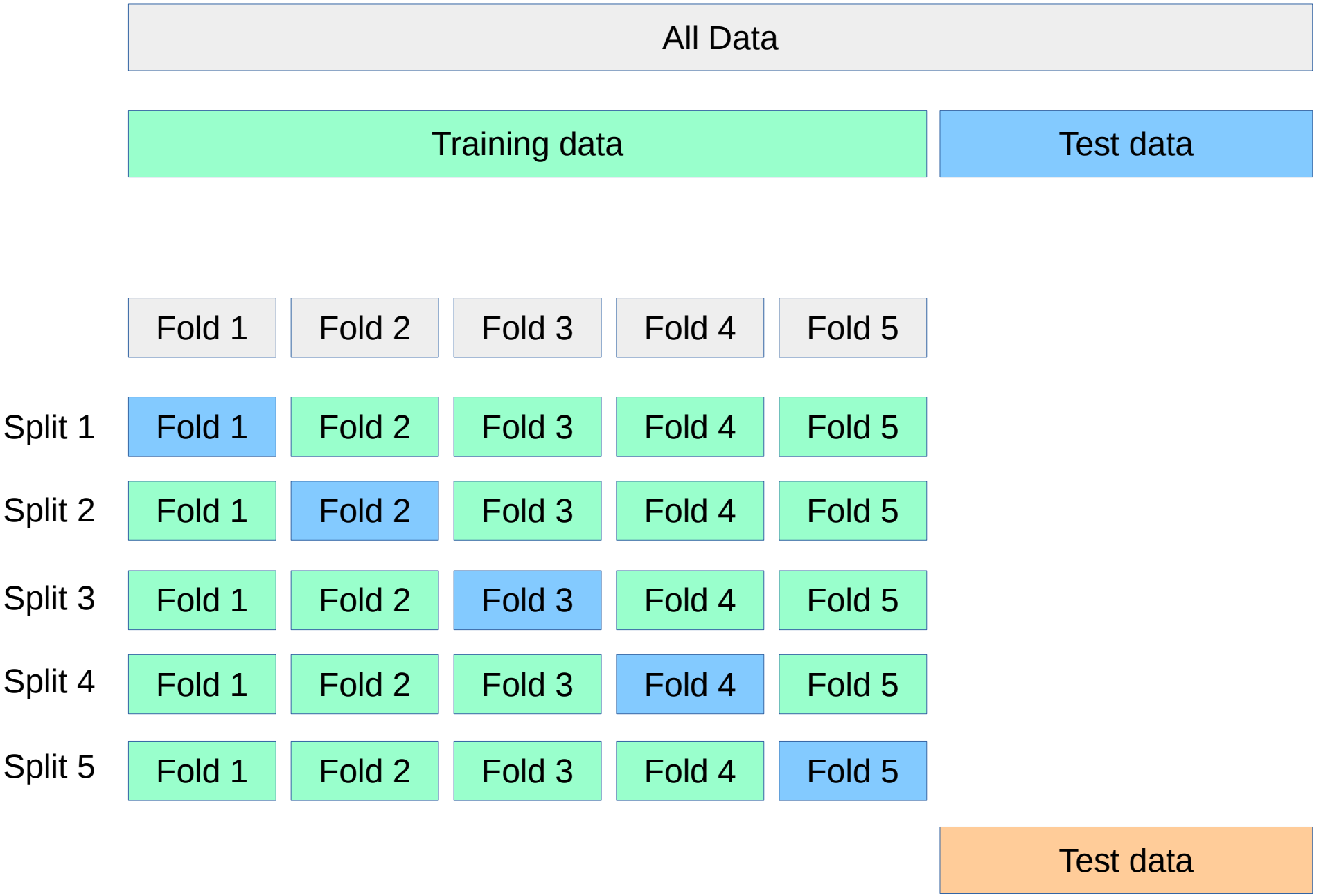

All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 2
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 3
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 4
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 5
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5

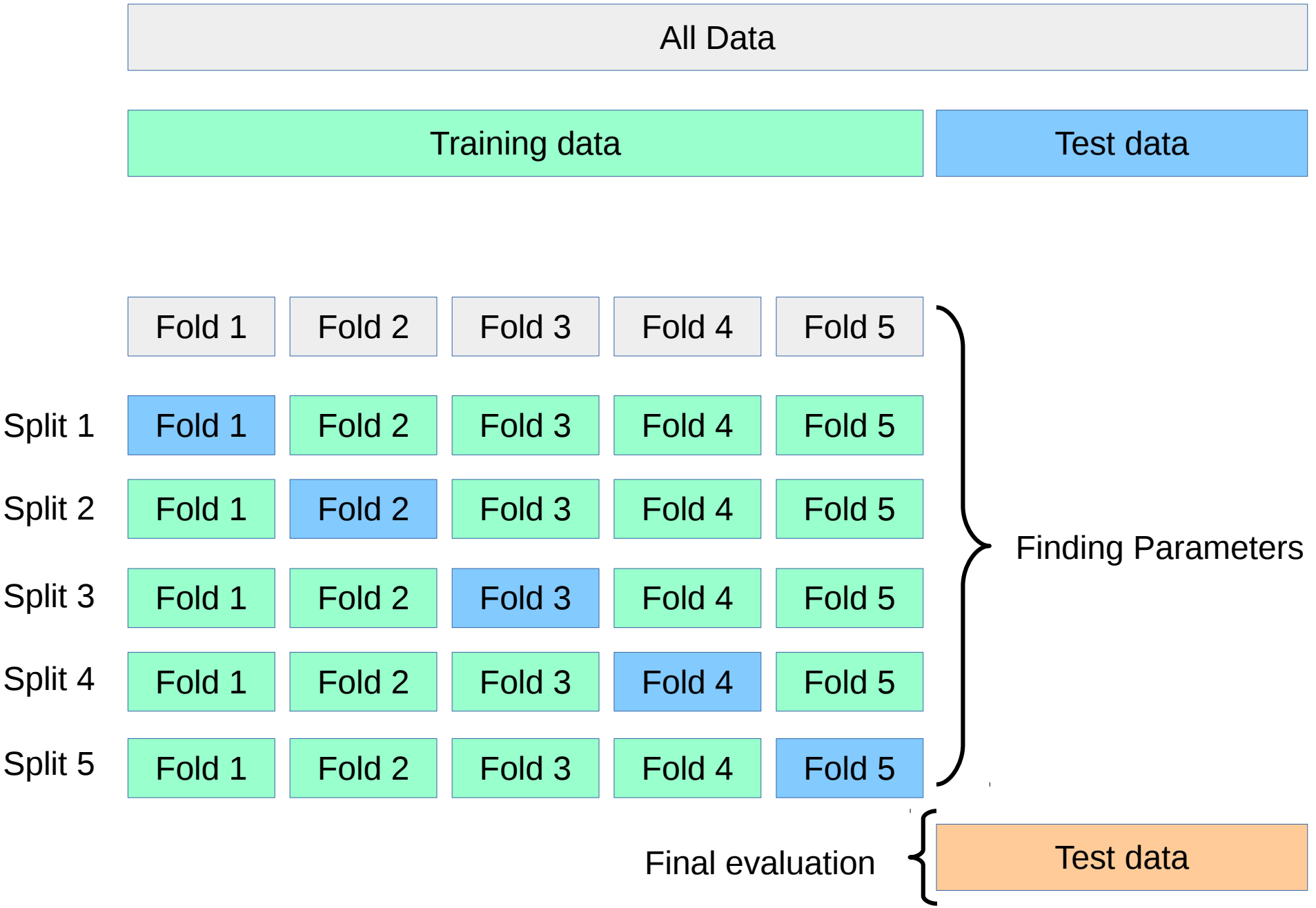
IPython Notebook:
Part 3 - Cross-validation

In [2]:
```
clf = SVC()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```


In [2]:
clf = SVC()
clf.fit(X_train, y_train)
SVC(self, C=1.0, kernel='rbf', degree=3, gamma=0.0, coef0=0.0,
shrinking=True, probability=False, tol=0.001, cache_size=200,
class_weight=None, verbose=False, max_iter=-1, random_state=N
one)


All Data
Training data
Test data


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Split 2
Split 3
Split 4
Split 5
Test data


All Data
Training data
Test data
Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Split 1
Split 2
Split 3
Split 4
Split 5
Finding Parameters
Final evaluation
Test data

SVC(C=0.001,
gamma=0.001)

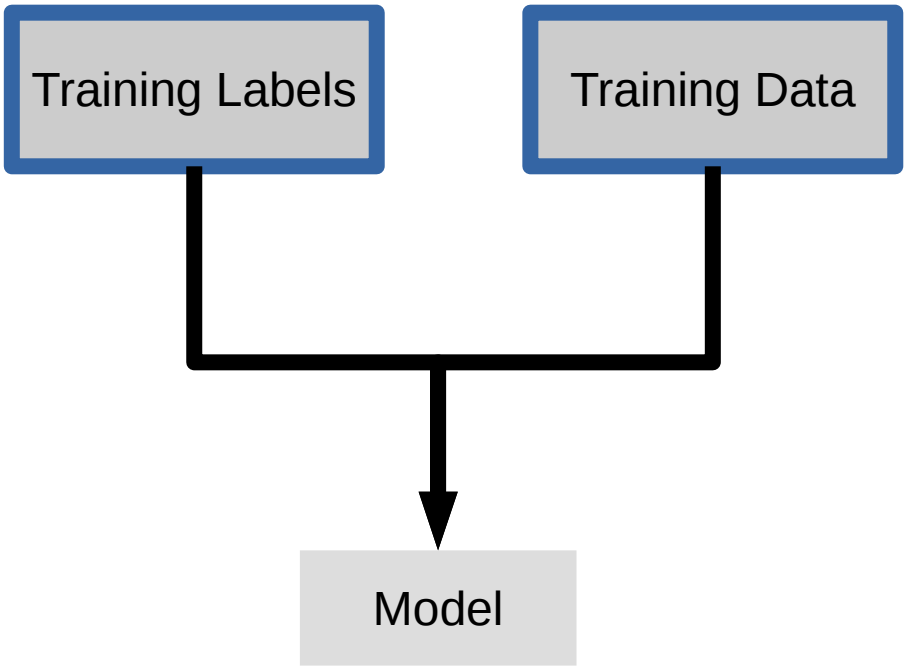
SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	SVC(C=0.1, gamma=0.001)	SVC(C=1, gamma=0.001)	SVC(C=10, gamma=0.001)

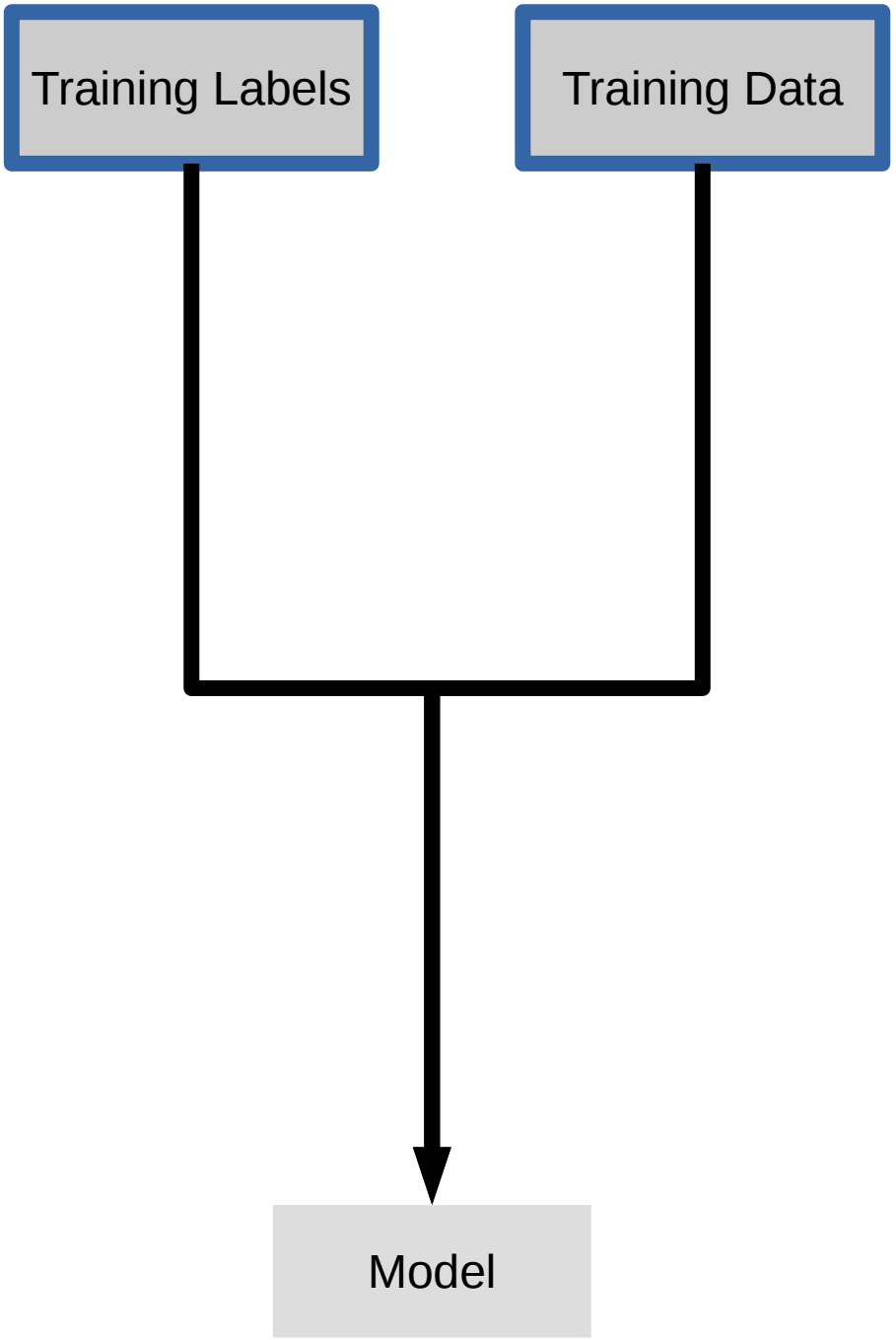
SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	SVC(C=0.1, gamma=0.001)	SVC(C=1, gamma=0.001)	SVC(C=10, gamma=0.001)
SVC(C=0.001, gamma=0.01)	SVC(C=0.01, gamma=0.01)	SVC(C=0.1, gamma=0.01)	SVC(C=1, gamma=0.01)	SVC(C=10, gamma=0.01)

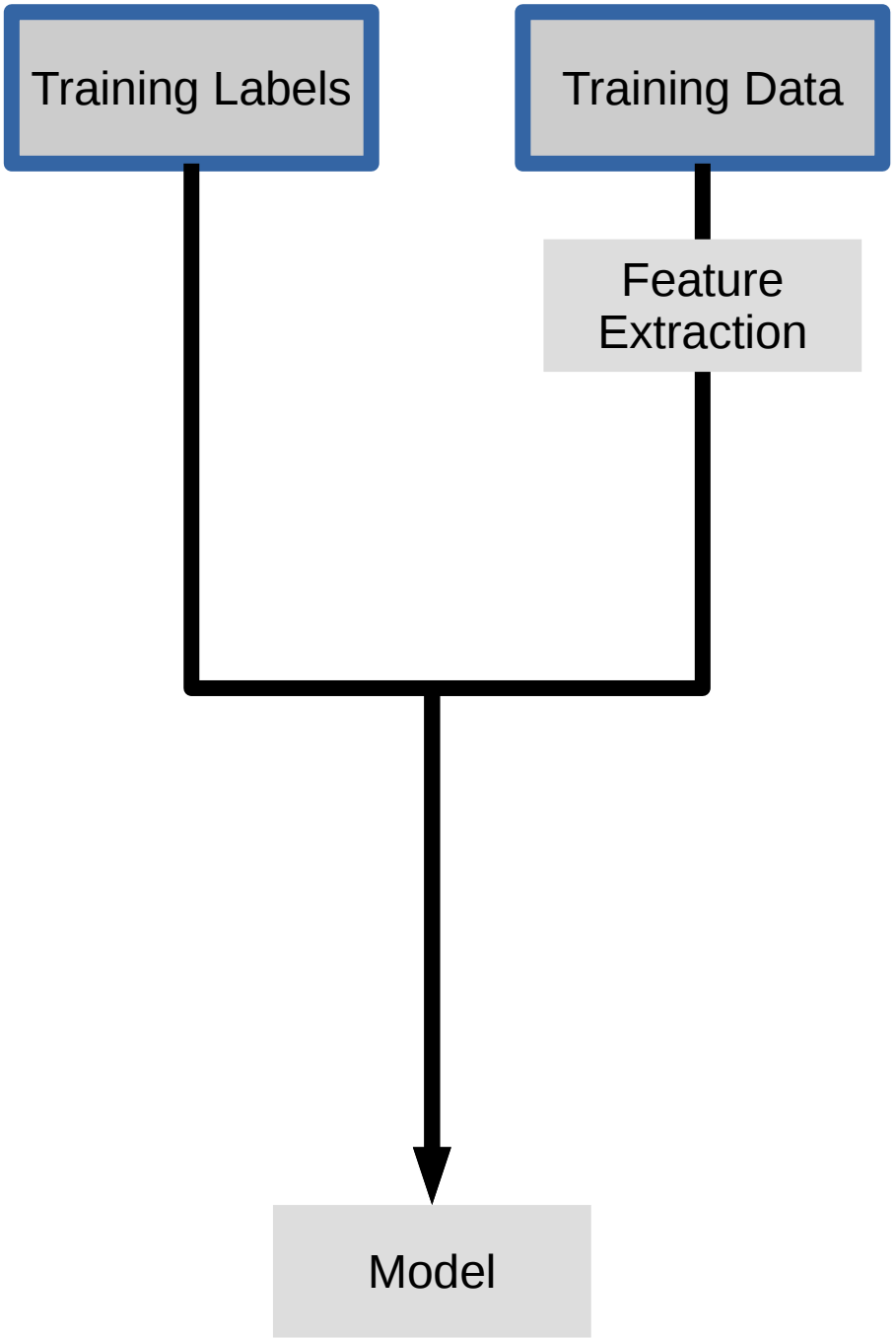
SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	SVC(C=0.1, gamma=0.001)	SVC(C=1, gamma=0.001)	SVC(C=10, gamma=0.001)
SVC(C=0.001, gamma=0.01)	SVC(C=0.01, gamma=0.01)	SVC(C=0.1, gamma=0.01)	SVC(C=1, gamma=0.01)	SVC(C=10, gamma=0.01)
SVC(C=0.001, gamma=0.1)	SVC(C=0.01, gamma=0.1)	SVC(C=0.1, gamma=0.1)	SVC(C=1, gamma=0.1)	SVC(C=10, gamma=0.1)

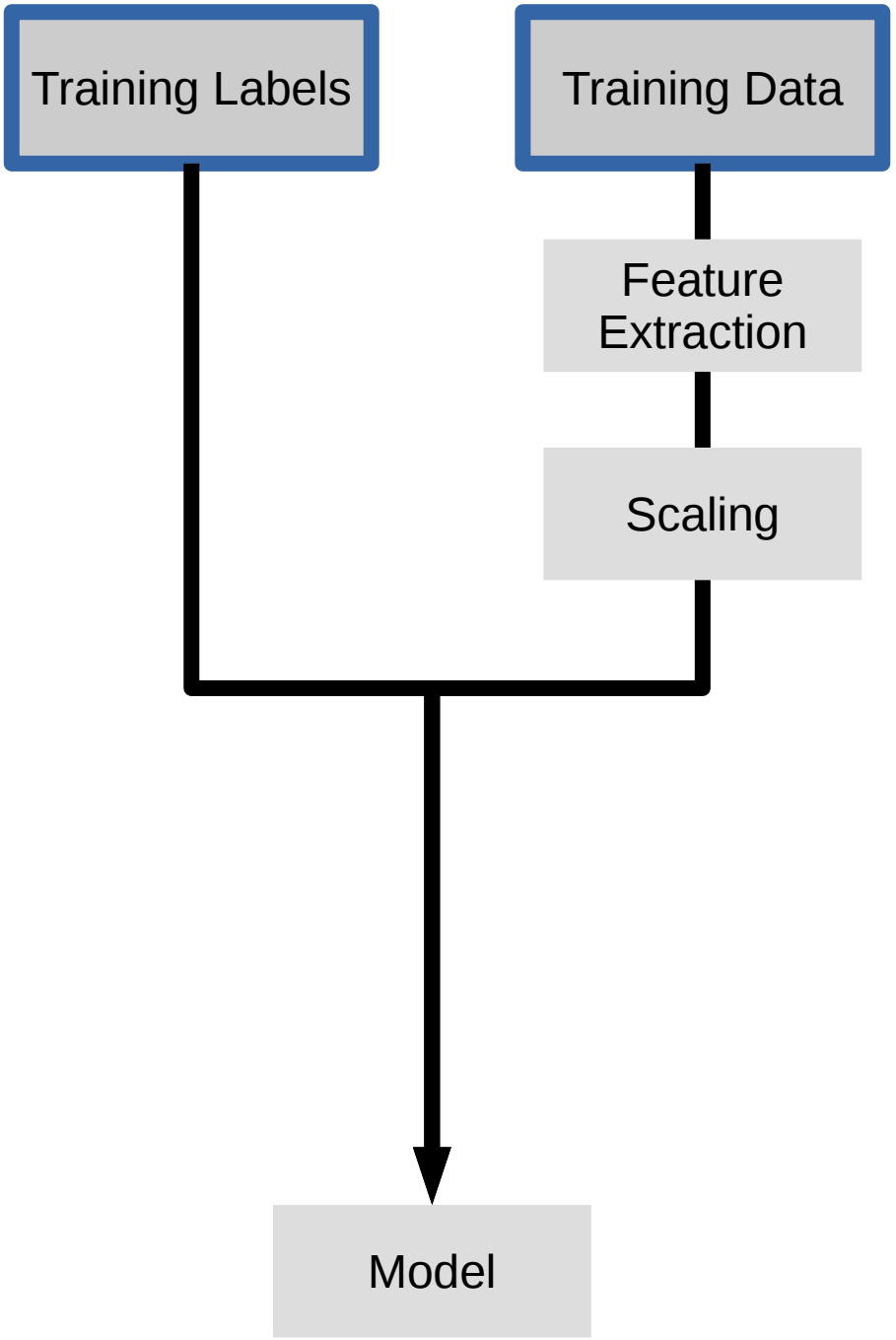
SVC(C=0.001, gamma=0.001)	SVC(C=0.01, gamma=0.001)	SVC(C=0.1, gamma=0.001)	SVC(C=1, gamma=0.001)	SVC(C=10, gamma=0.001)
SVC(C=0.001, gamma=0.01)	SVC(C=0.01, gamma=0.01)	SVC(C=0.1, gamma=0.01)	SVC(C=1, gamma=0.01)	SVC(C=10, gamma=0.01)
SVC(C=0.001, gamma=0.1)	SVC(C=0.01, gamma=0.1)	SVC(C=0.1, gamma=0.1)	SVC(C=1, gamma=0.1)	SVC(C=10, gamma=0.1)
SVC(C=0.001, gamma=1)	SVC(C=0.01, gamma=1)	SVC(C=0.1, gamma=1)	SVC(C=1, gamma=1)	SVC(C=10, gamma=1)
SVC(C=0.001, gamma=10)	SVC(C=0.01, gamma=10)	SVC(C=0.1, gamma=10)	SVC(C=1, gamma=10)	SVC(C=10, gamma=10)

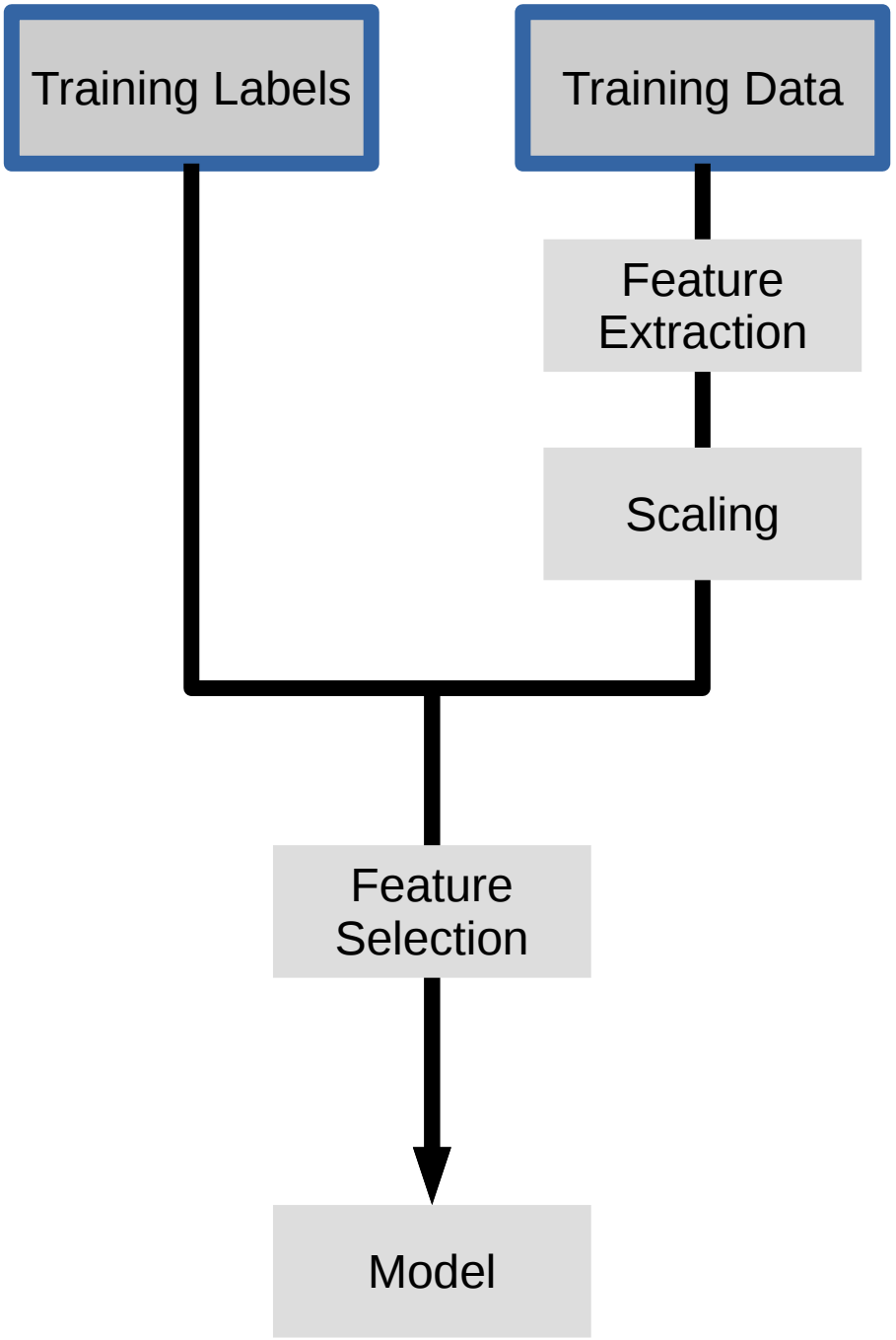
IPython Notebook:
Part 4 – Grid Searches

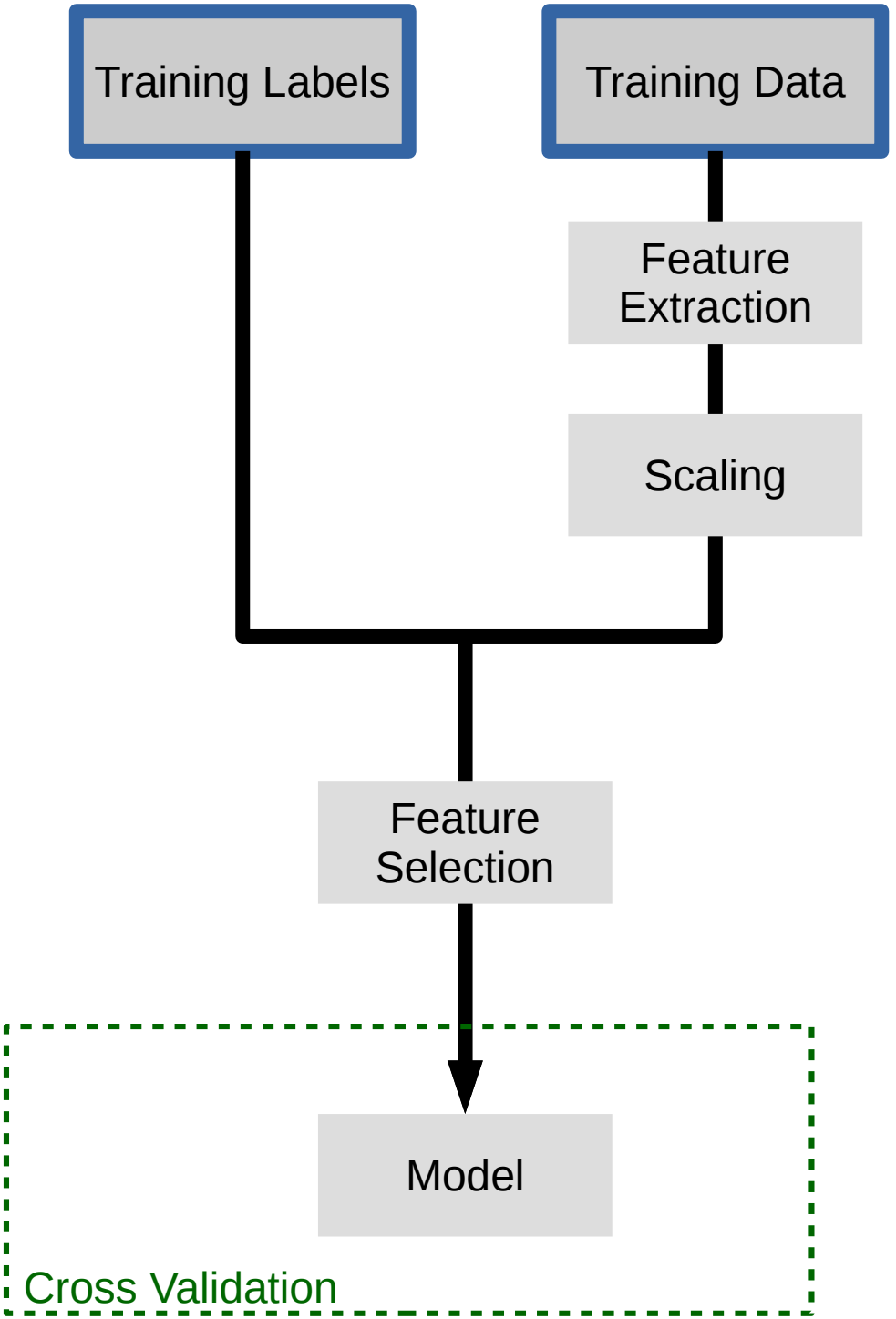

Training Labels
Training Data
Model

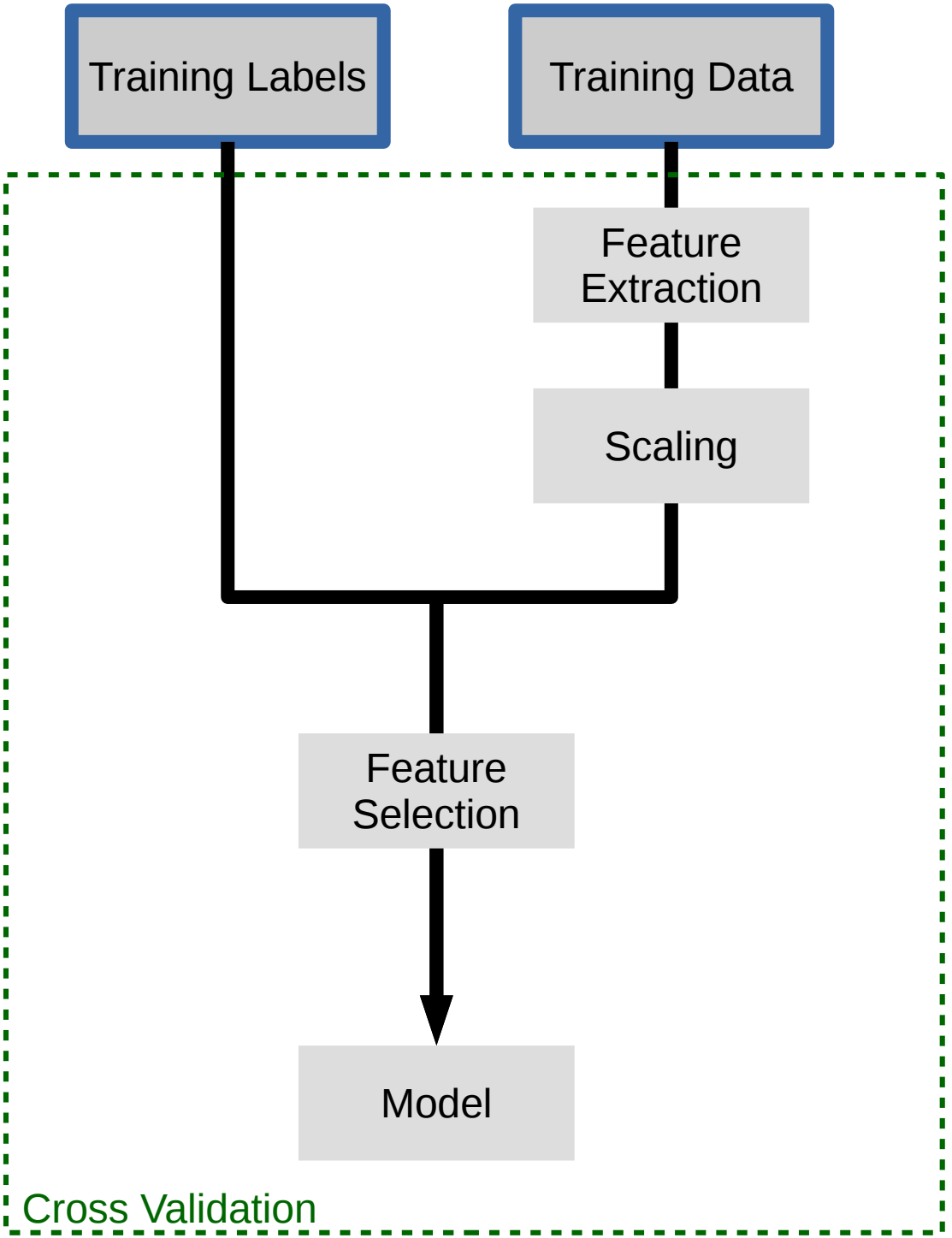

Training Labels
Training Data
Model


Training Labels
Training Data
Feature Extraction
Model


Training Labels
Training Data
Feature Extraction
Scaling
Model


Training Labels
Training Data
Feature Extraction
Scaling
Feature Selection
Model


Training Labels
Training Data
Feature Extraction
Scaling
Feature Selection
Model
Cross Validation


Training Labels
Training Data
Feature Extraction
Scaling
Feature Selection
Model
Cross Validation

IPython Notebook:
Part 5 - Preprocessing and Pipelines

Do cross-validation over all steps jointly.
Keep a separate test set until the very end.

Bag Of Word Representations

`CountVectorizer / TfidfVectorizer`

Bag Of Word Representations

`CountVectorizer / TfidfVectorizer`

`“This is how you get ants.”`

Bag Of Word Representations

`CountVectorizer / TfidfVectorizer`

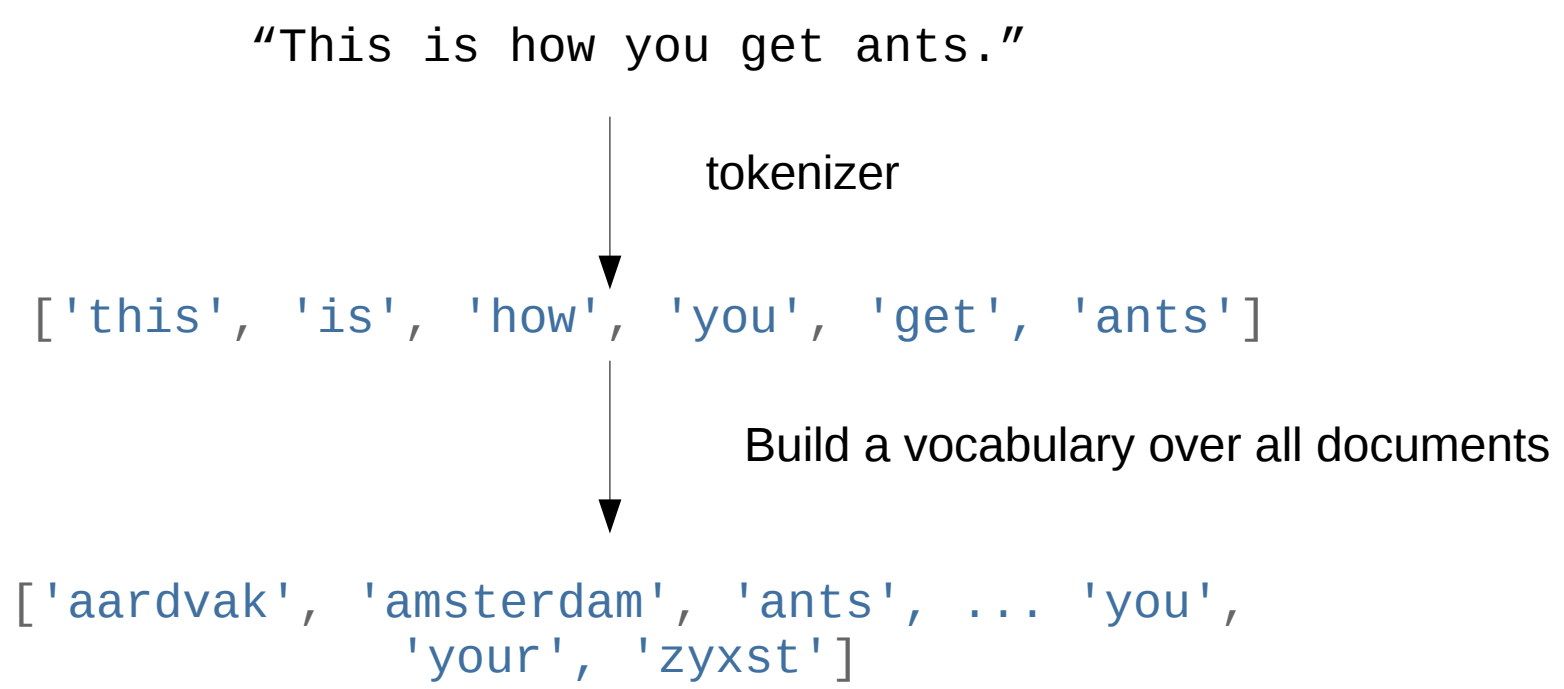
`“This is how you get ants.”`

tokenizer

```
['this', 'is', 'how', 'you', 'get', 'ants']
```

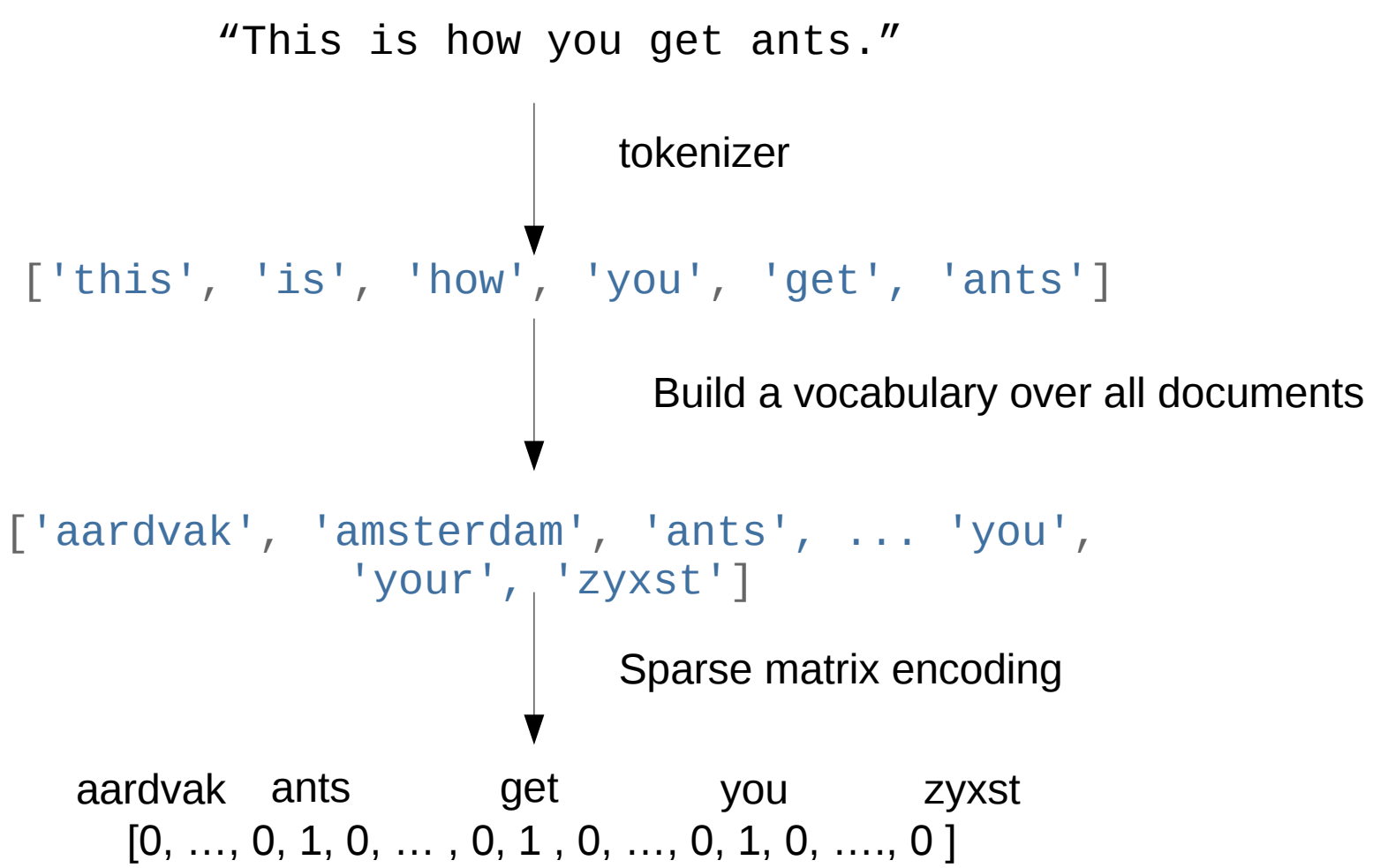
Bag Of Word Representations

`CountVectorizer / TfidfVectorizer`



Bag Of Word Representations

`CountVectorizer / TfidfVectorizer`



Application: Insult detection

Application: Insult detection

i really don't understand your point. It seems that you are mixing apples and oranges.

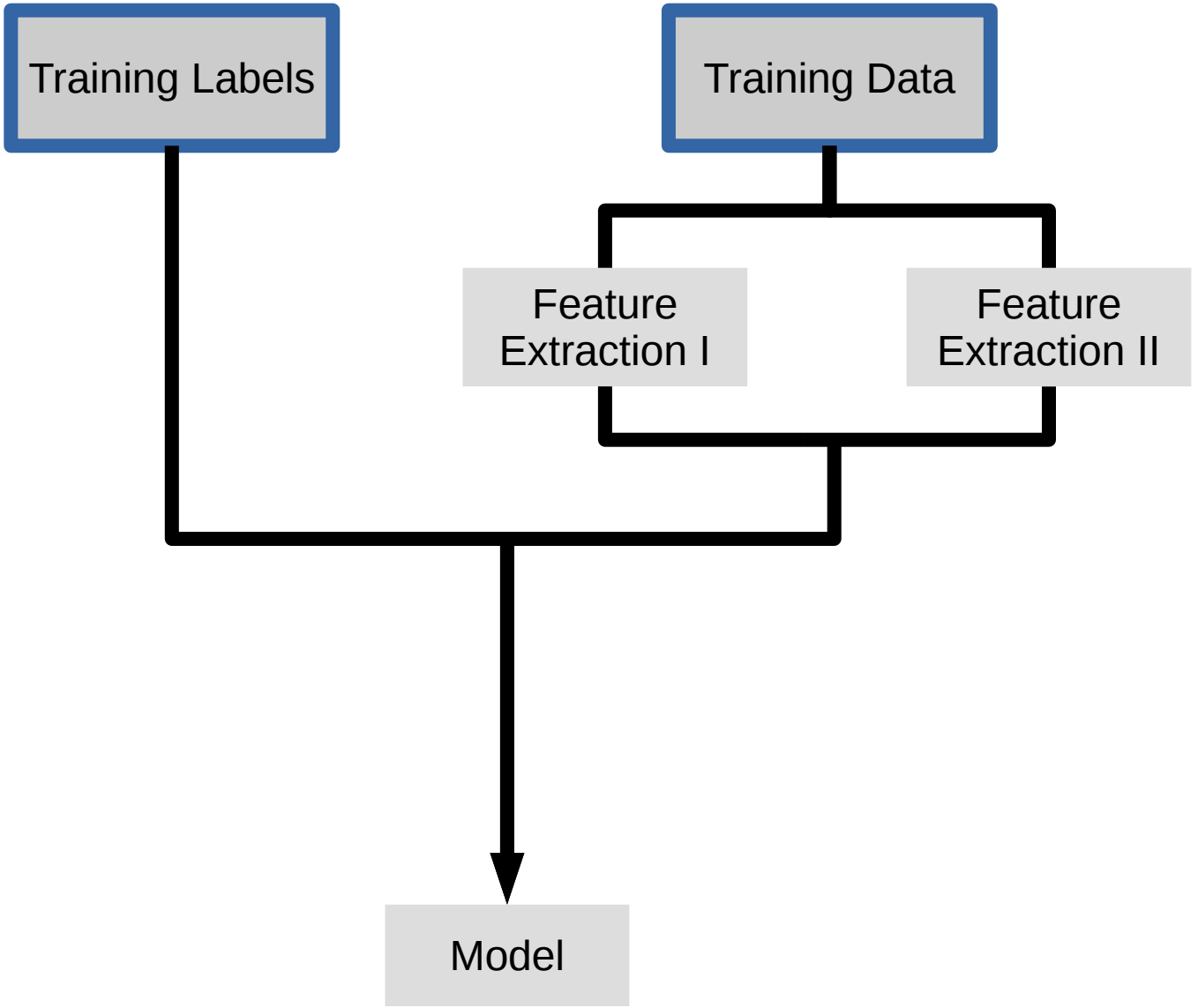
Application: Insult detection

i really don't understand your point. It seems that you are mixing apples and oranges.

Clearly you're a fucktard.

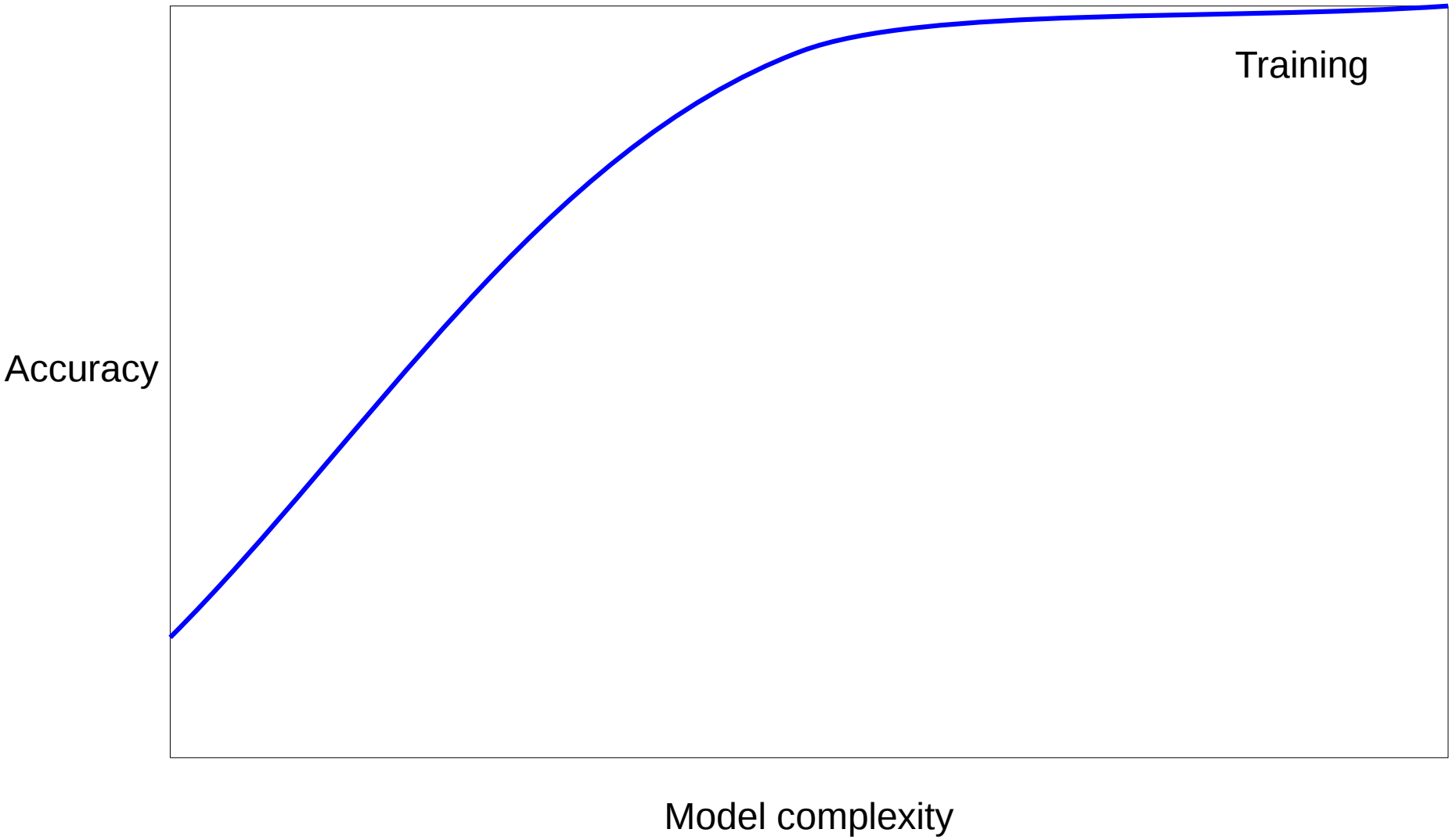
IPython Notebook:
Part 6 - Working With Text Data

Feature Union

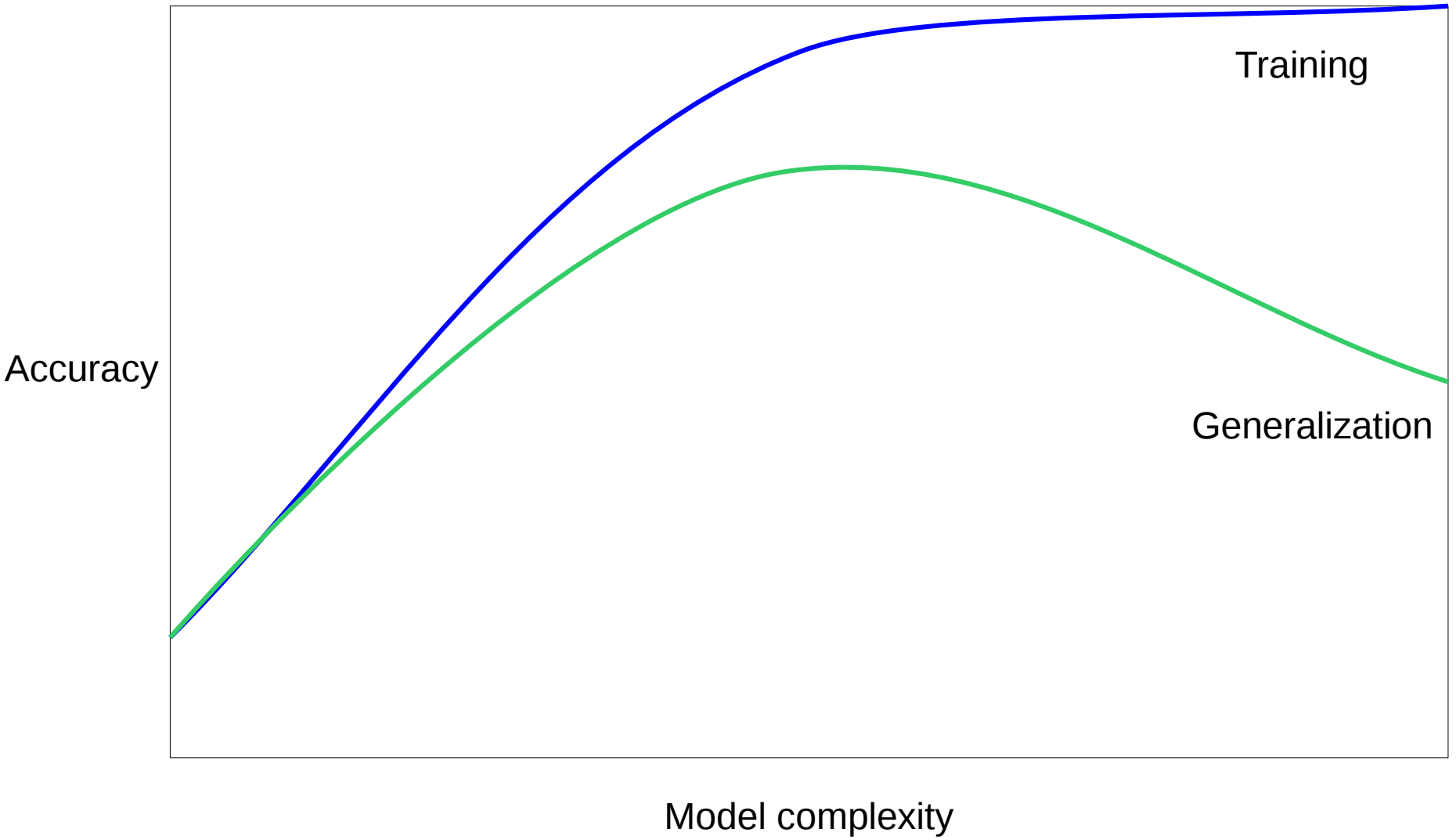


IPython Notebook:
Part 7 – FeatureUnion

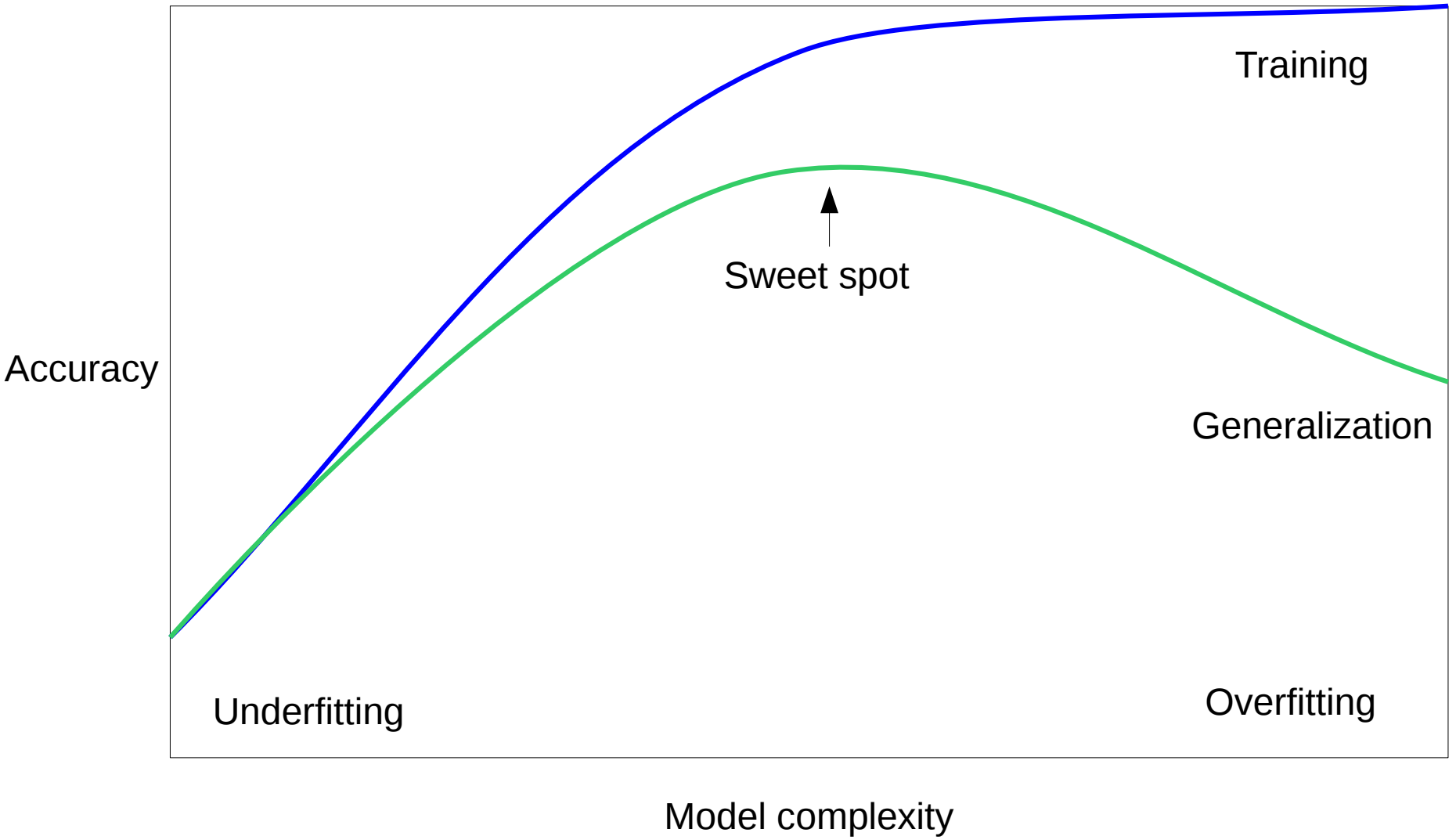
Overfitting and Underfitting



Overfitting and Underfitting



Overfitting and Underfitting

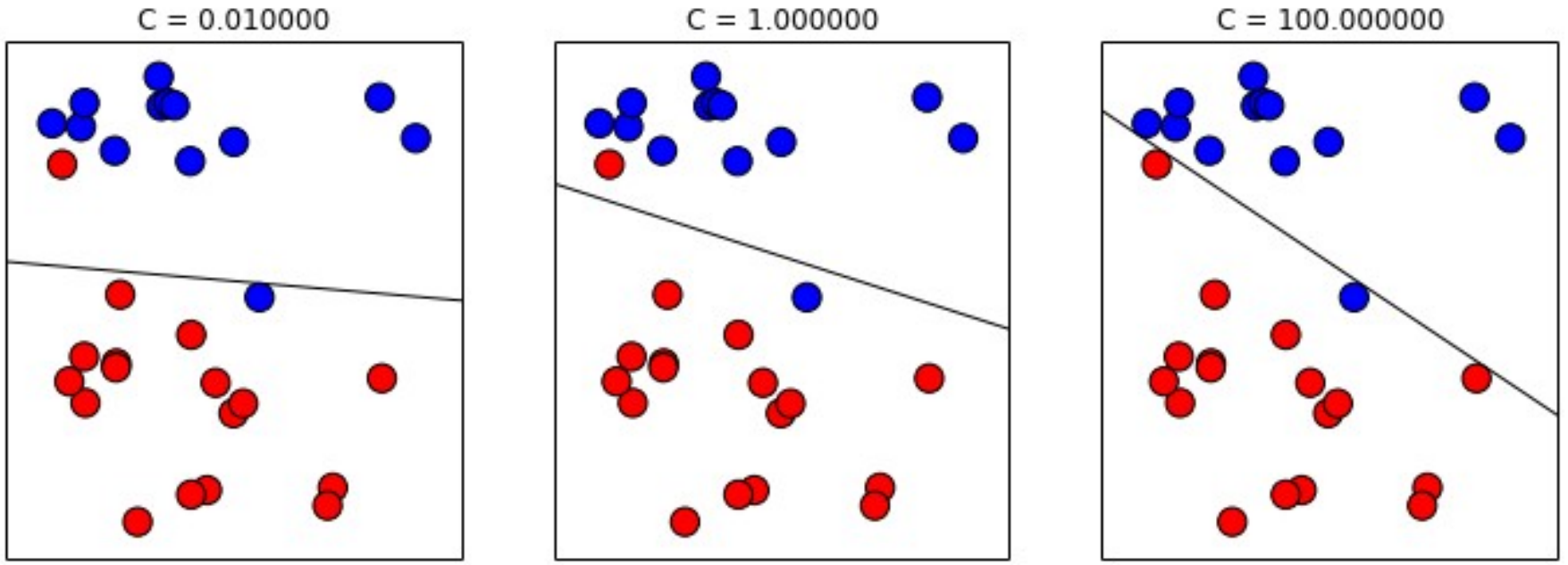


Linear SVM

$$\hat{y} = \text{sign}(w_0 + \sum_i w_i x_i)$$

Linear SVM

$$\hat{y} = \text{sign}(w_0 + \sum_i w_i x_i)$$



(RBF) Kernel SVM

$$\hat{y} = \text{sign}(\alpha_0 + \sum_j \alpha_j y_j k(\mathbf{x^{(j)}}, \mathbf{x}))$$

(RBF) Kernel SVM

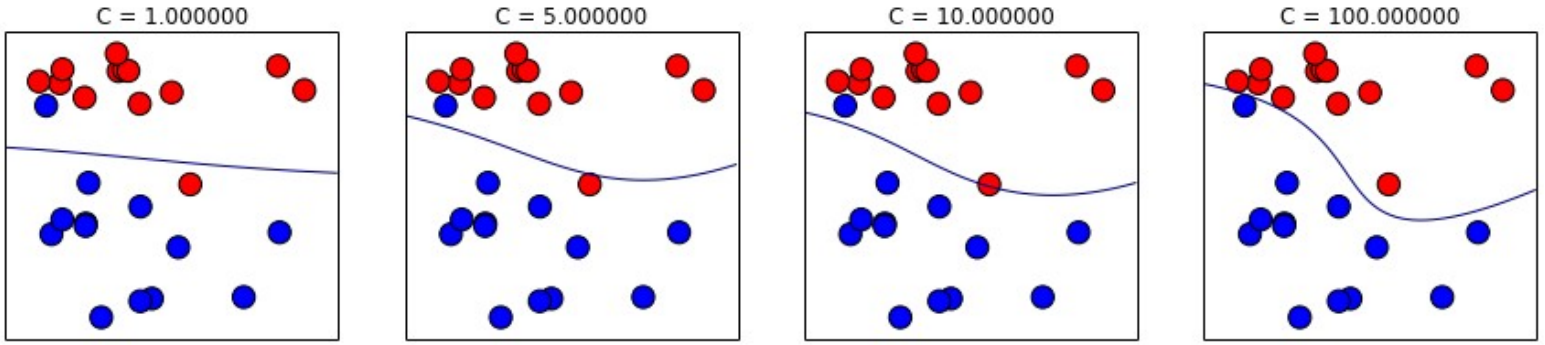
$$\hat{y} = \text{sign}(\alpha_0 + \sum_j \alpha_j y_j k(\mathbf{x^{(j)}}, \mathbf{x}))$$

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2)$$

(RBF) Kernel SVM

$$\hat{y} = \text{sign}(\alpha_0 + \sum_j \alpha_j y_j k(\mathbf{x}^{(\mathbf{j})}, \mathbf{x}))$$

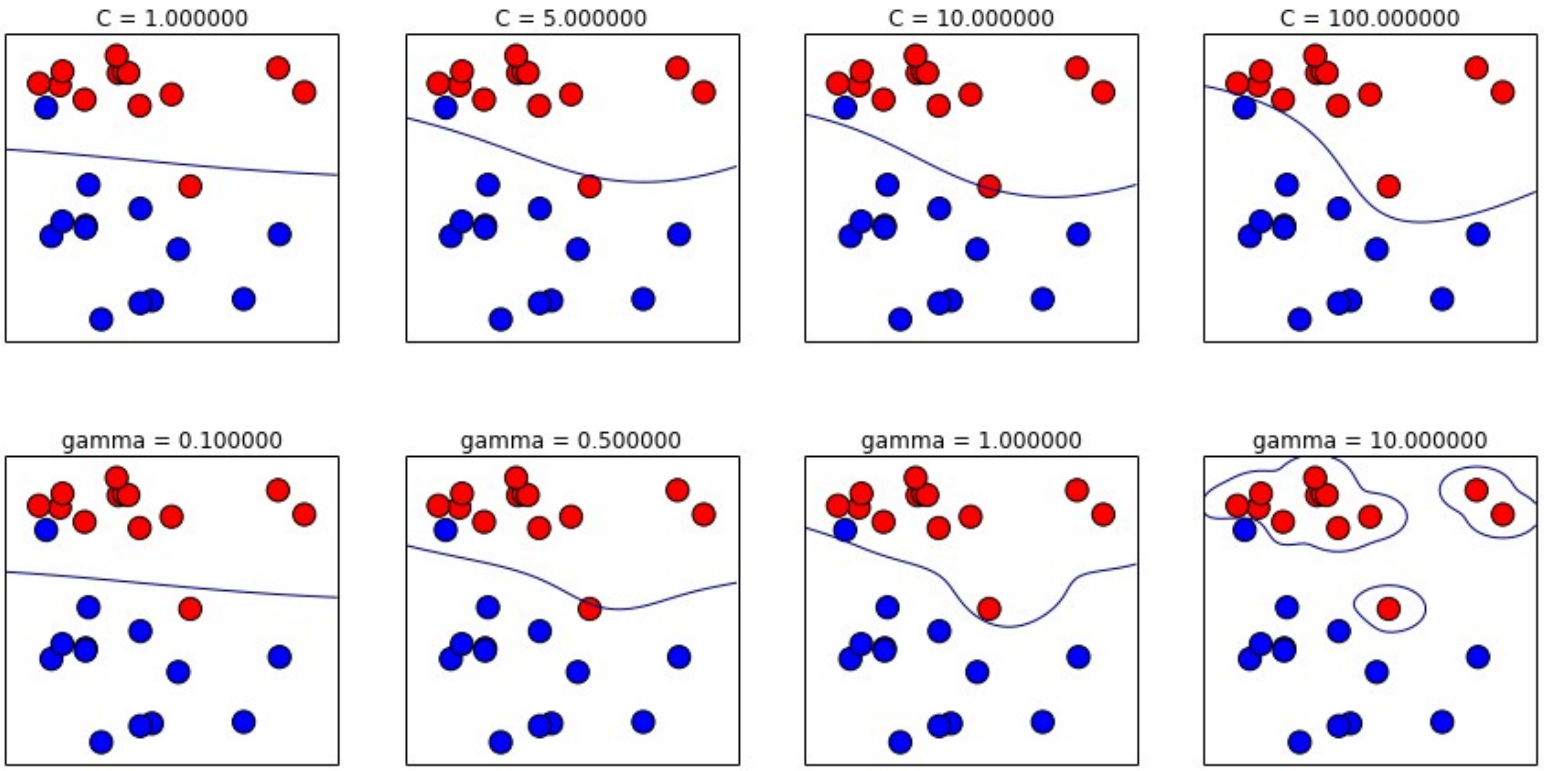
$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2)$$



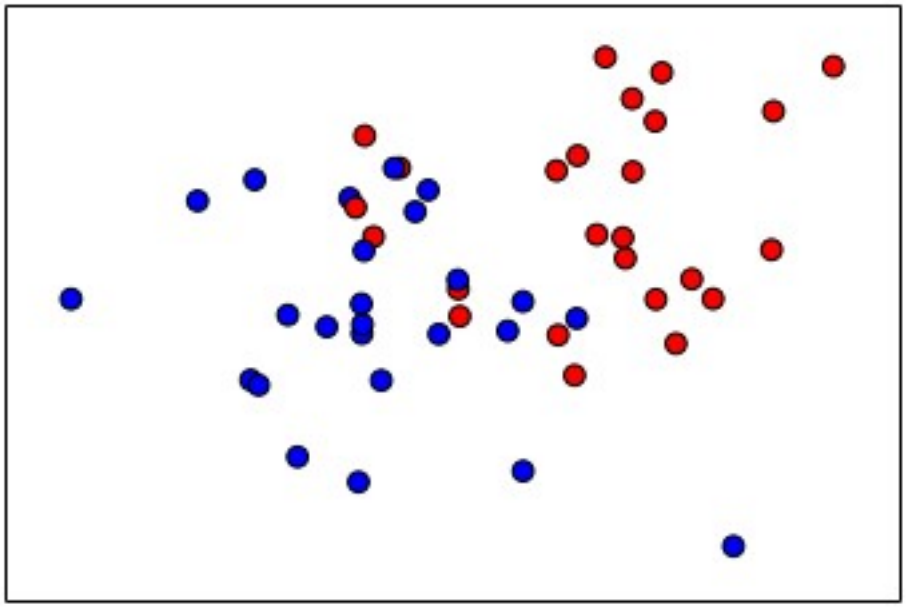
(RBF) Kernel SVM

$$\hat{y} = \text{sign}(\alpha_0 + \sum_j \alpha_j y_j k(\mathbf{x}^{(\mathbf{j})}, \mathbf{x}))$$

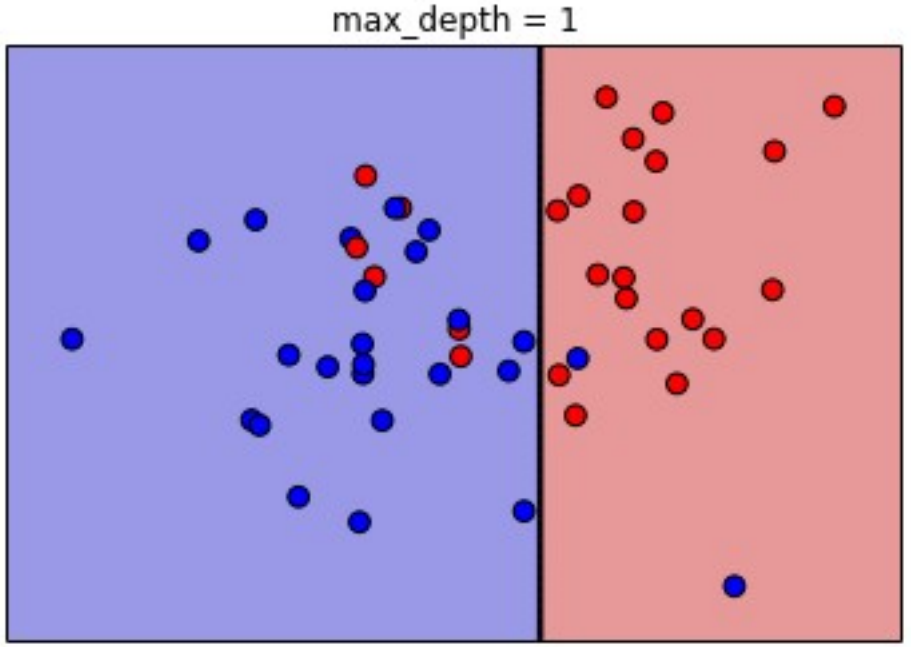
$$k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2)$$

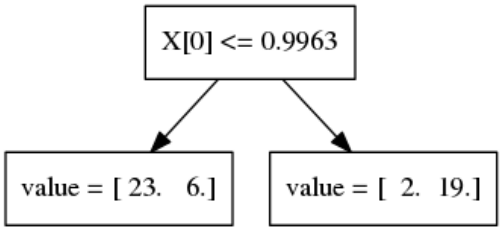


Decision Trees

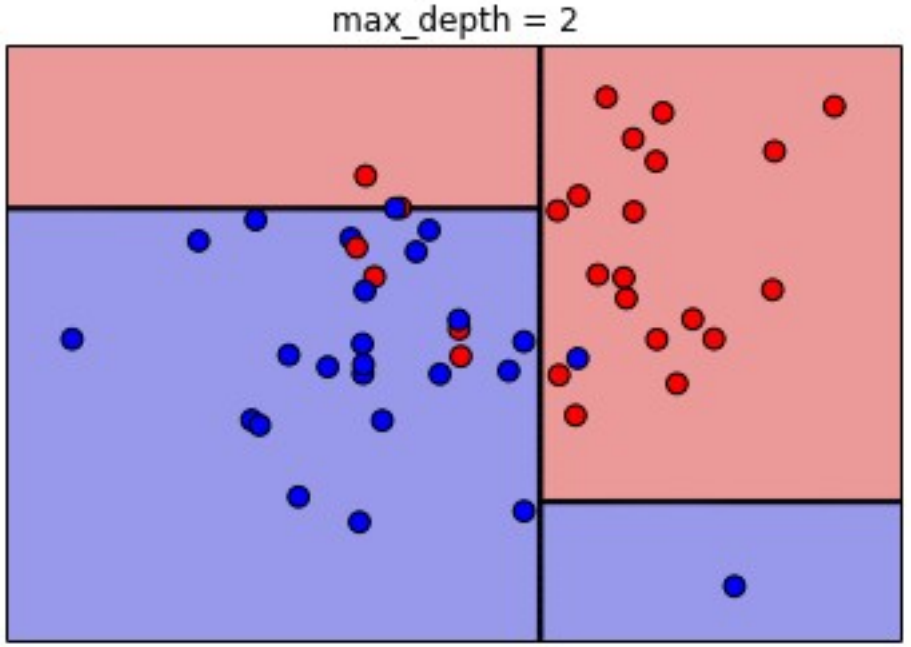


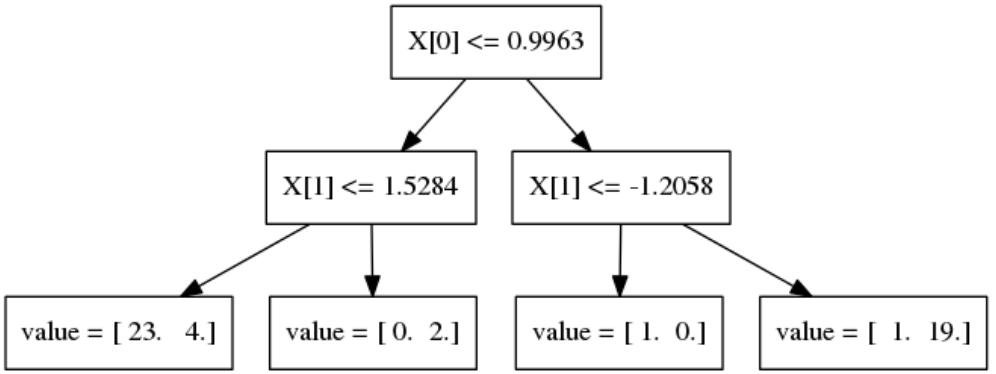
Decision Trees



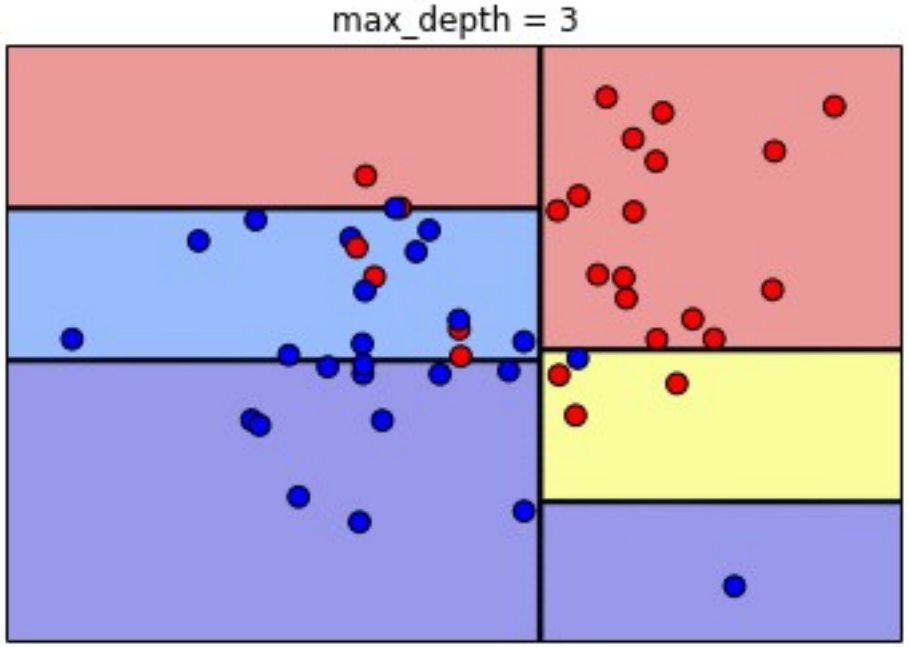


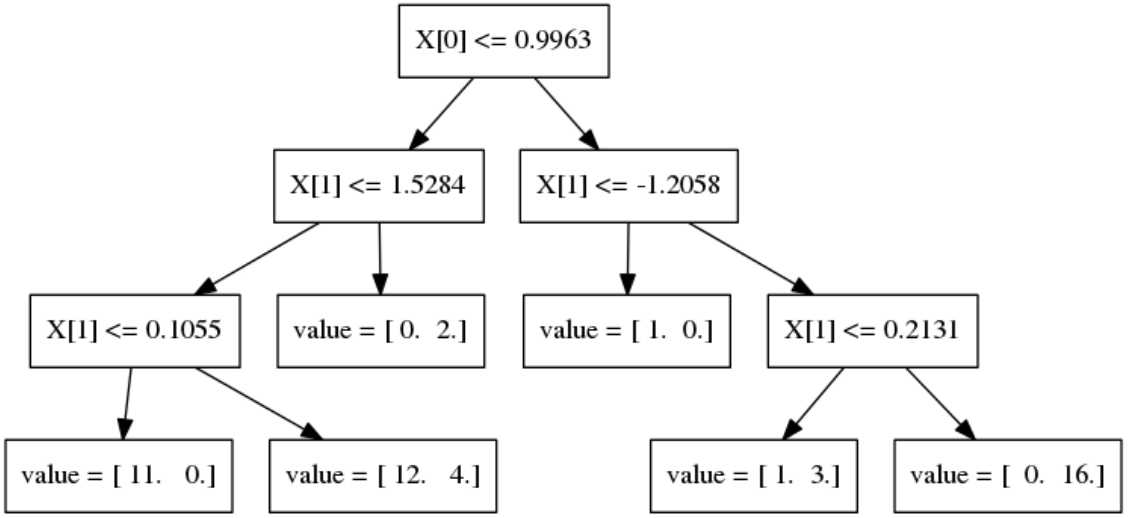
Decision Trees



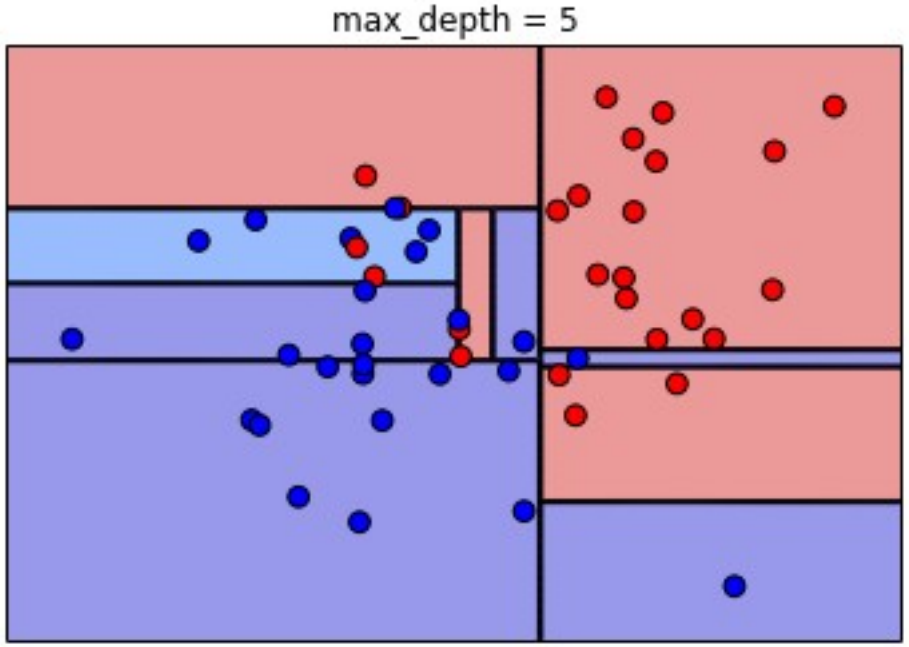


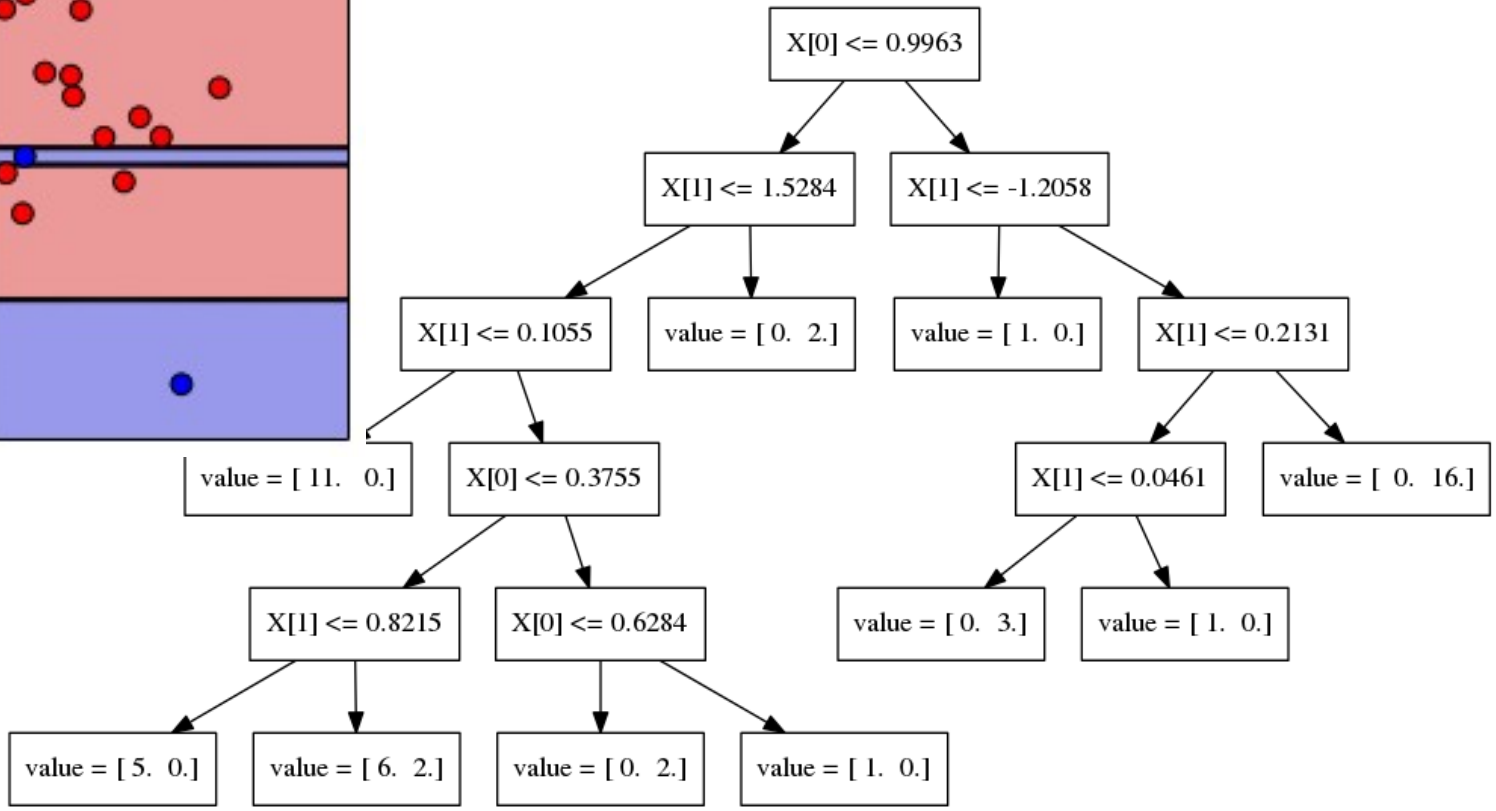
Decision Trees



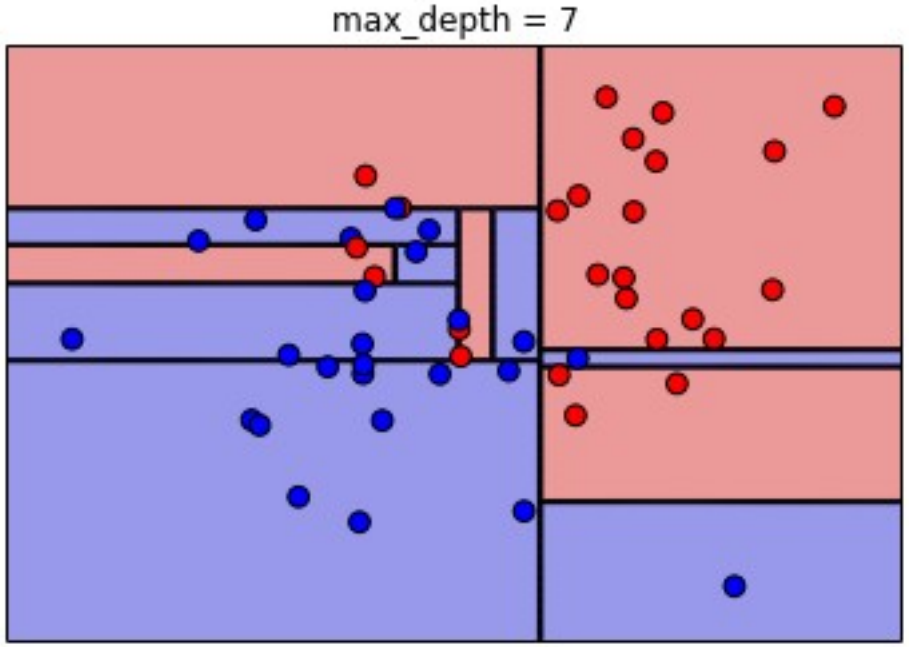


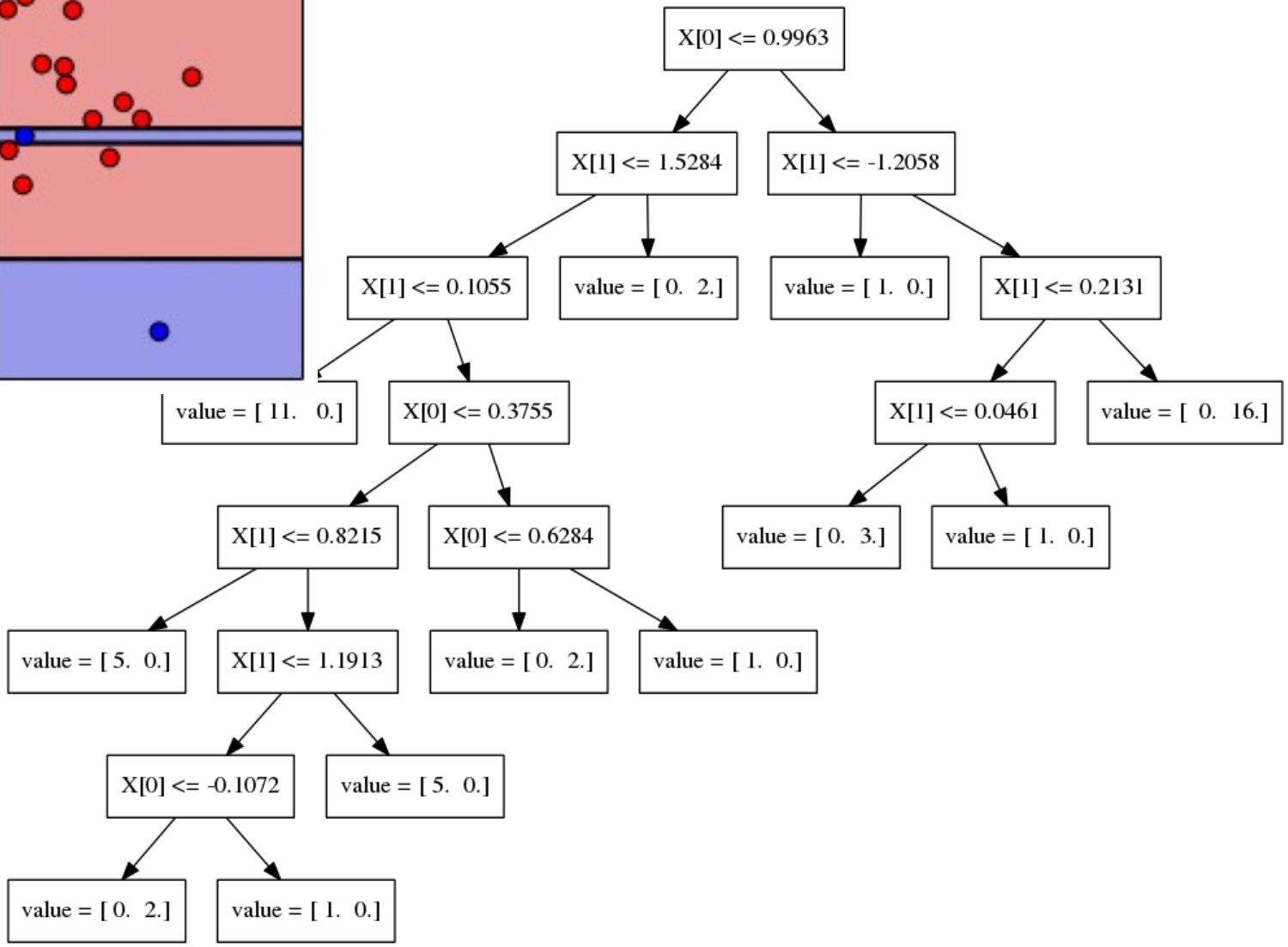
Decision Trees



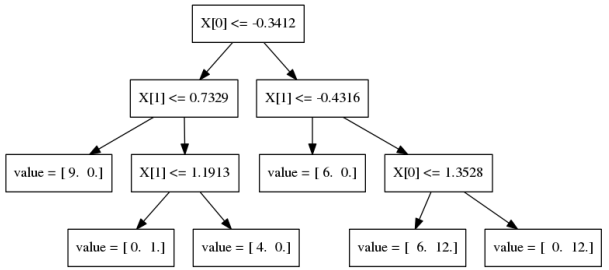


Decision Trees

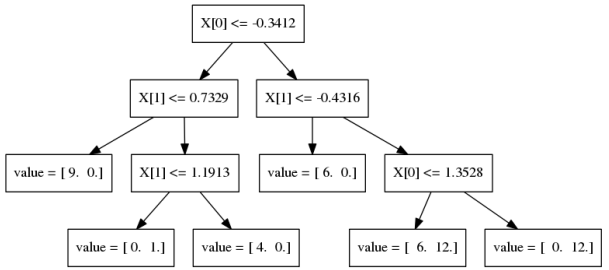


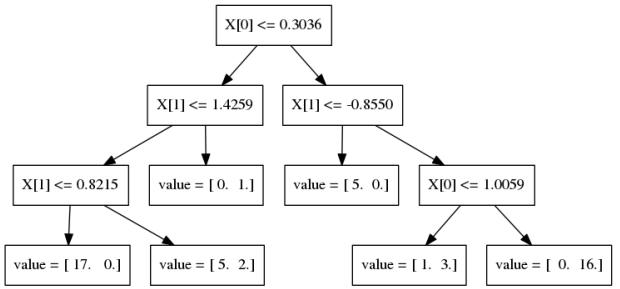


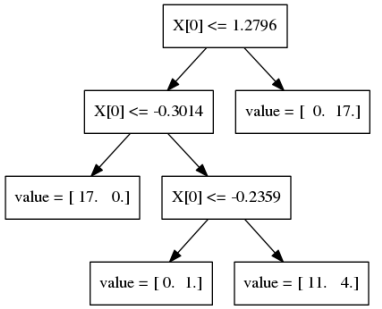
Random Forests



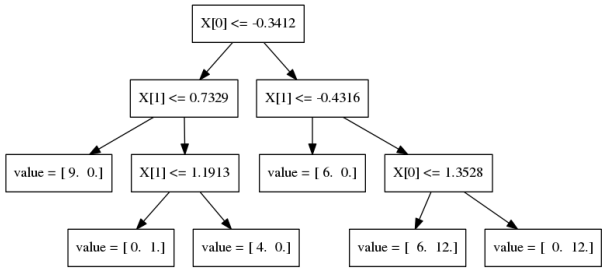
Random Forests

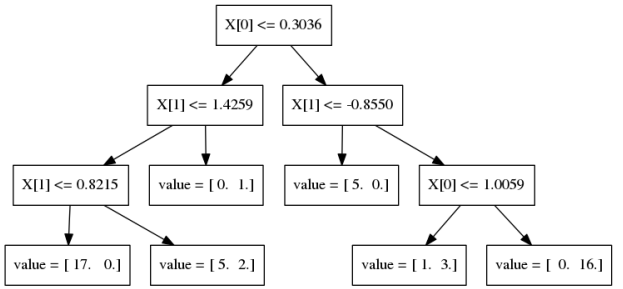


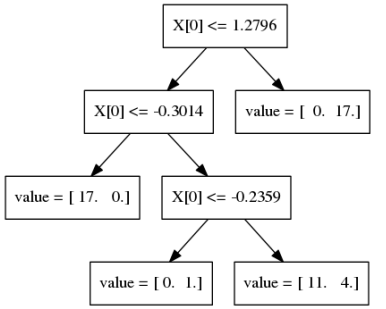


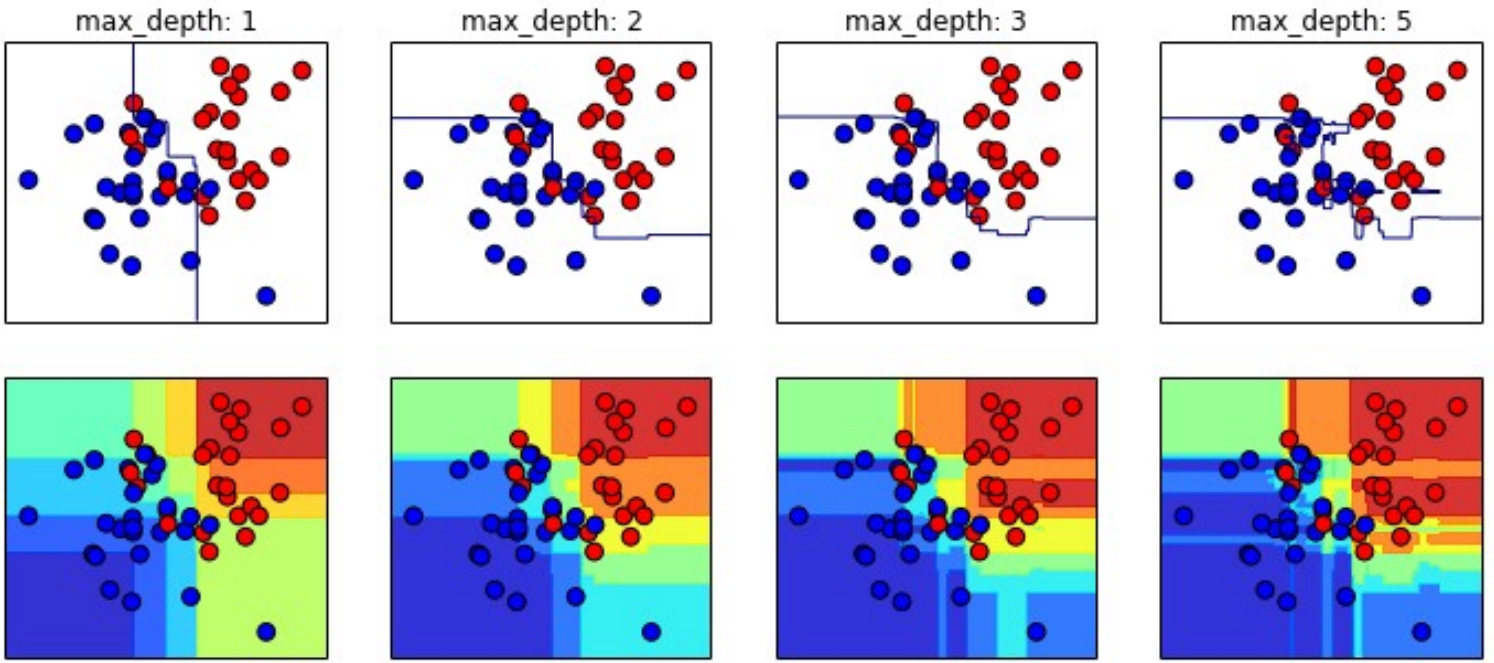


Random Forests



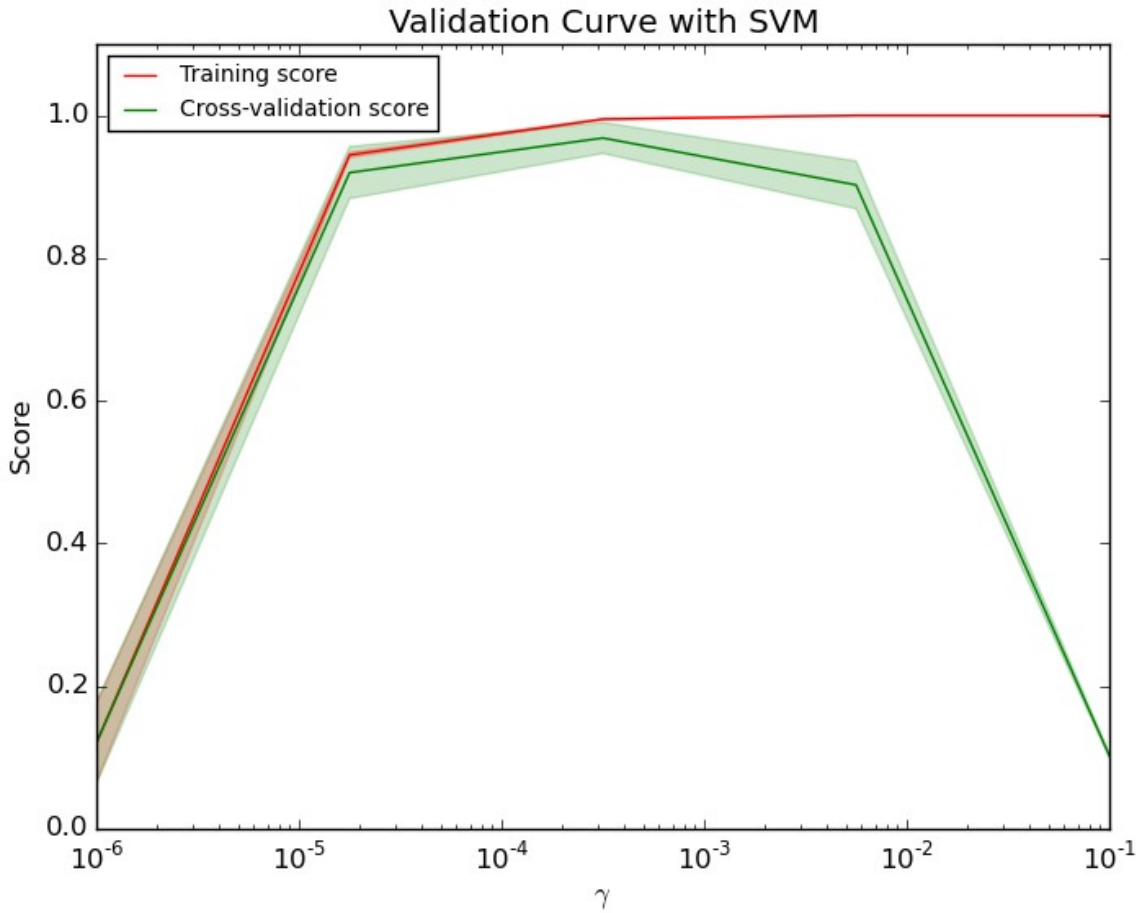






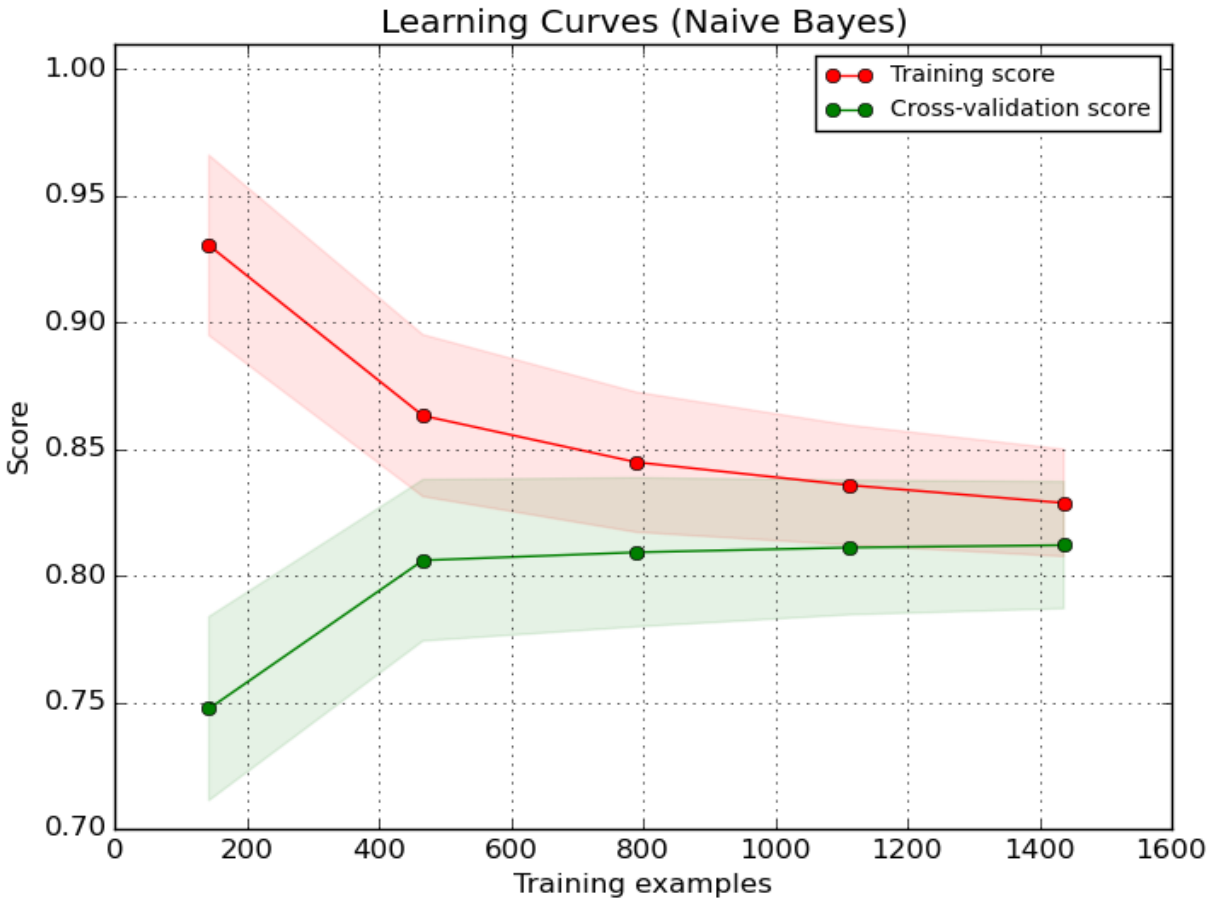
Validation Curves

```
train_scores, test_scores = validation_curve(SVC(), X, y,
    param_name="gamma", param_range=param_range)
```



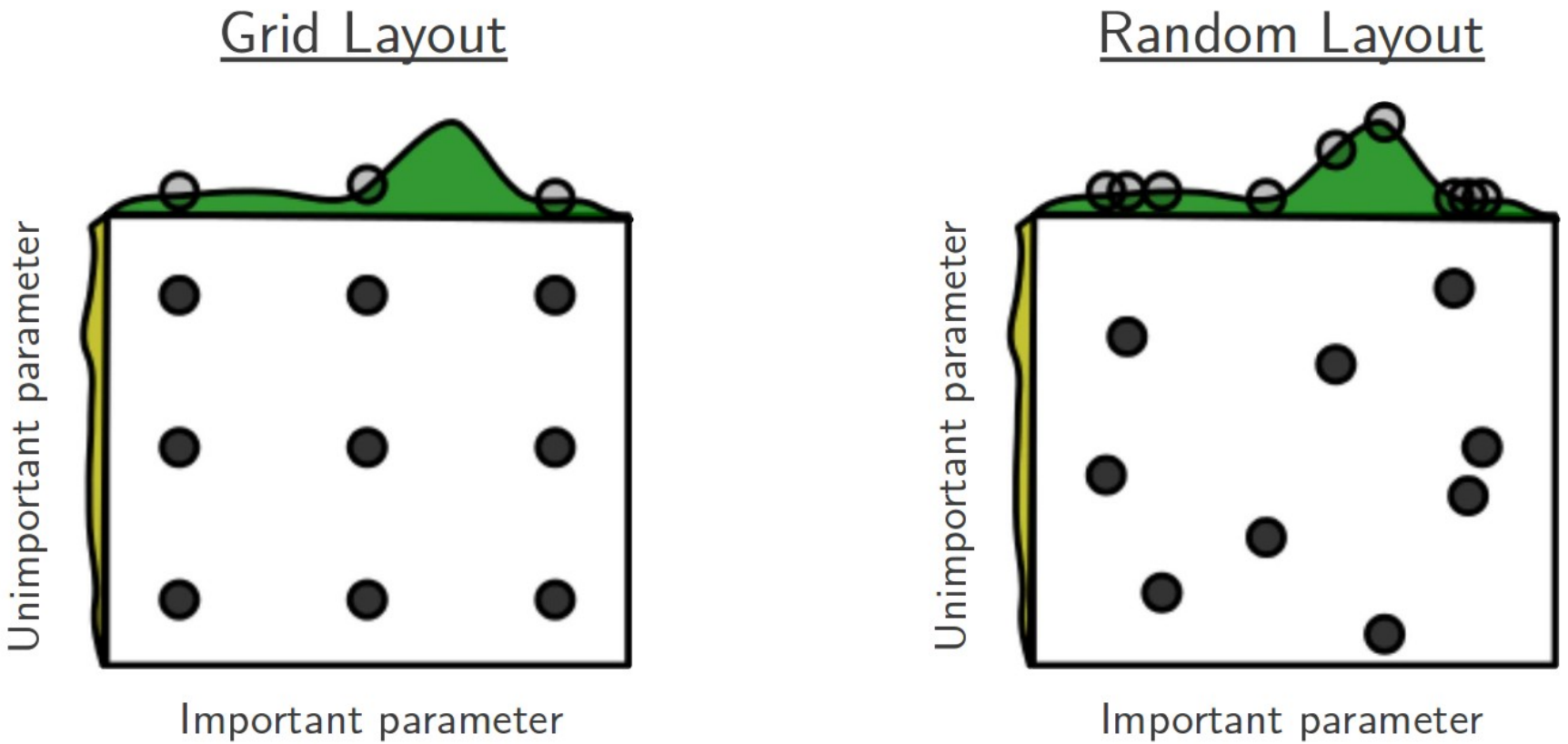
Learning Curves

```
train_sizes, train_scores, test_scores = learning_curve(
    estimator, X, y,train_sizes=train_sizes)
```



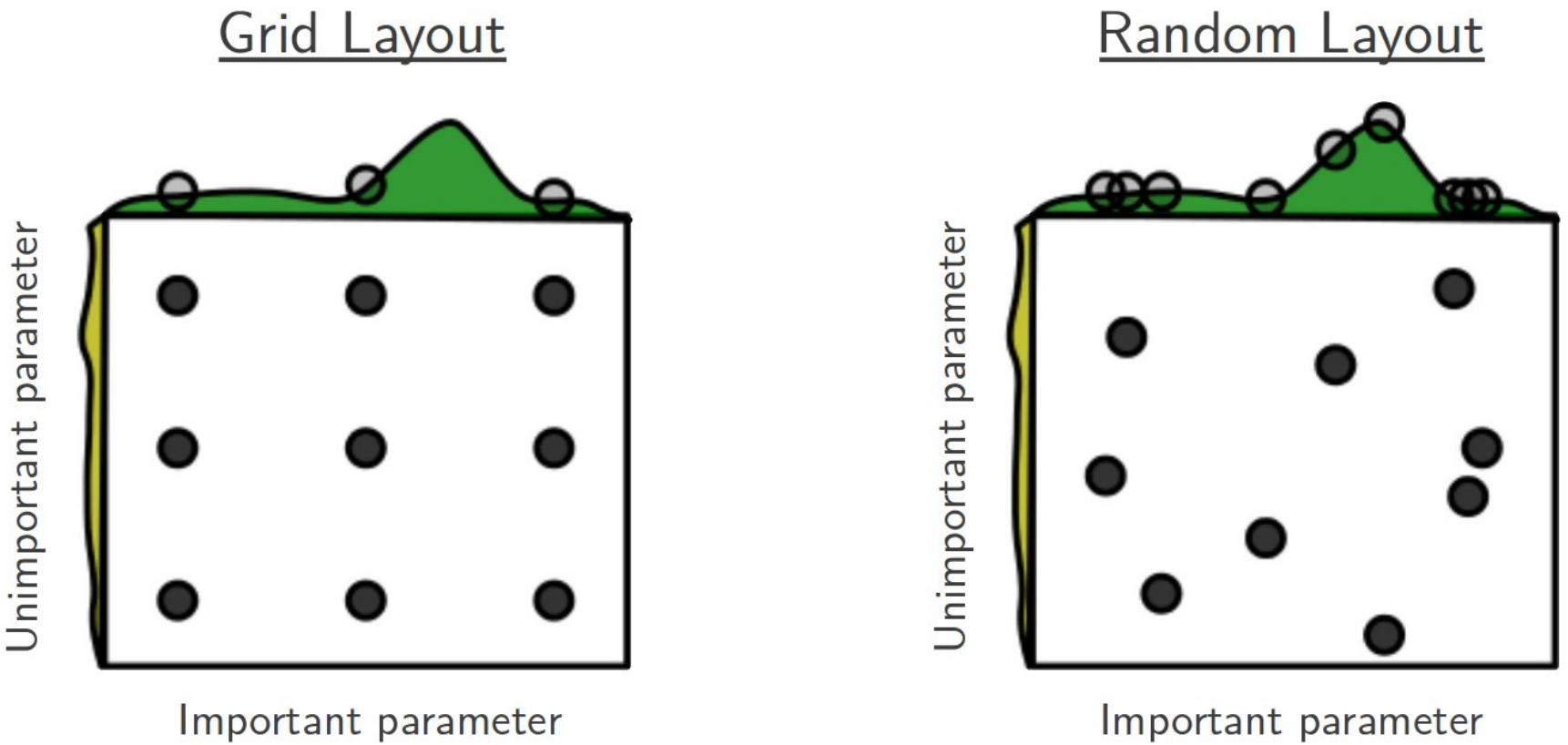
Randomized Parameter Search

Randomized Parameter Search



Source: Bergstra and Bengio

Randomized Parameter Search



Step-size free for continuous parameters
Decouples runtime from search-space size
Robust against irrelevant parameters

Source: Bergstra and Bengio

Randomized Parameter Search

- Always use distributions for continuous variables.
- Don't use for low dimensional spaces.
- Future: Bayesian optimization based search.

IPython Notebook:
Part 8 – Randomized Parameter Search

Generalized Cross-Validation and Path Algorithms

IPython Notebook:
Part 9 – Generalized Cross-Validation

Linear Models	Feature Selection	Tree-Based models [possible]
LogisticRegressionCV [new]	RFECV	[DecisionTreeCV]
RidgeCV		[RandomForestClassifierCV]
RidgeClassifierCV		[GradientBoostingClassifierCV]
LarsCV		
ElasticNetCV		
...		

Scoring Functions

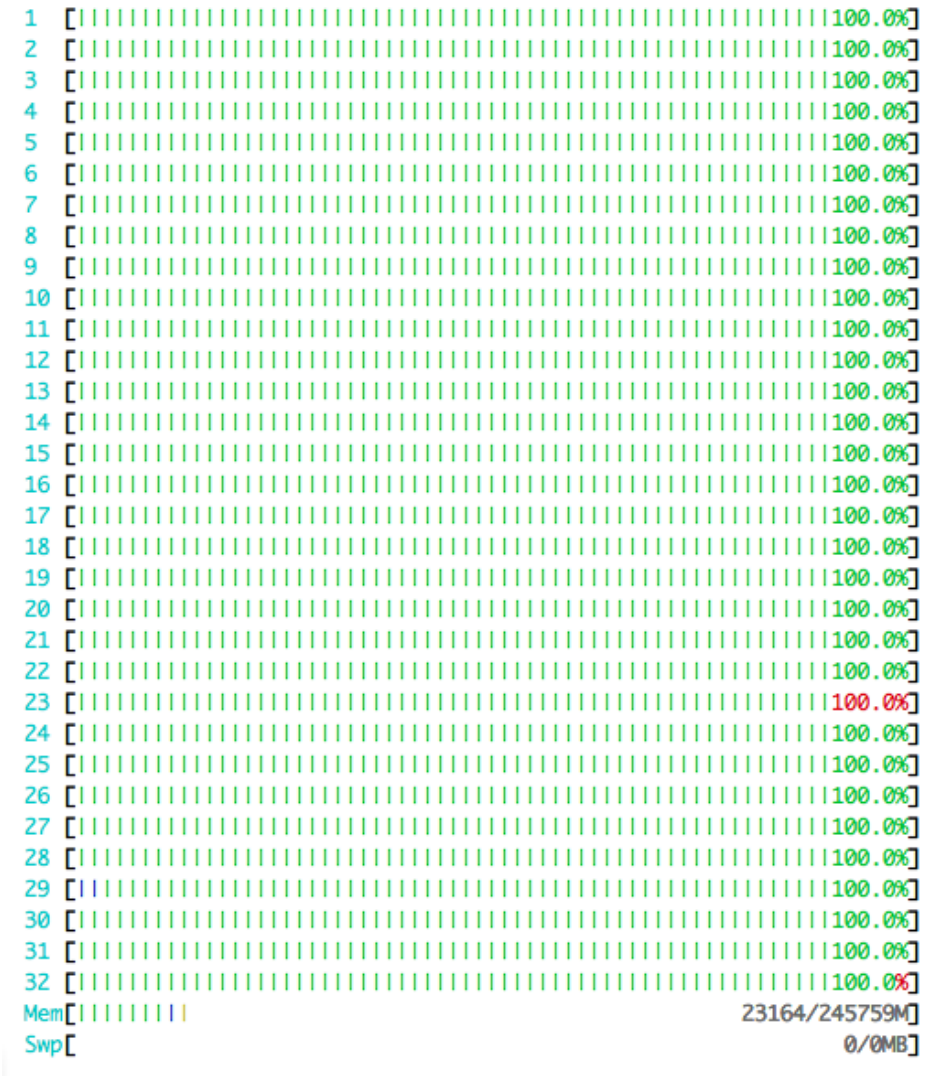
GridSeachCV
RandomizedSearchCV
cross_val_score
...CV

Default:
Accuracy (classification)
R2 (regression)

IPython Notebook:
Part 10 – Scoring Metrics

Out of Core Learning

Or: save ourself the effort



Think twice!

- Old laptop: 4GB Ram
- 1073741824 float32
- Or 1mio data points with 1000 features
- EC2 : 256 GB Ram
- 68719476736 float32
- Or 68mio data points with 1000 features

Supported Algorithms

- All `SGDClassifier` derivatives
- Naive Bayes
- `MinibatchKMeans`
- `Birch`
- `IncrementalPCA`
- `MiniBatchDictionaryLearning`

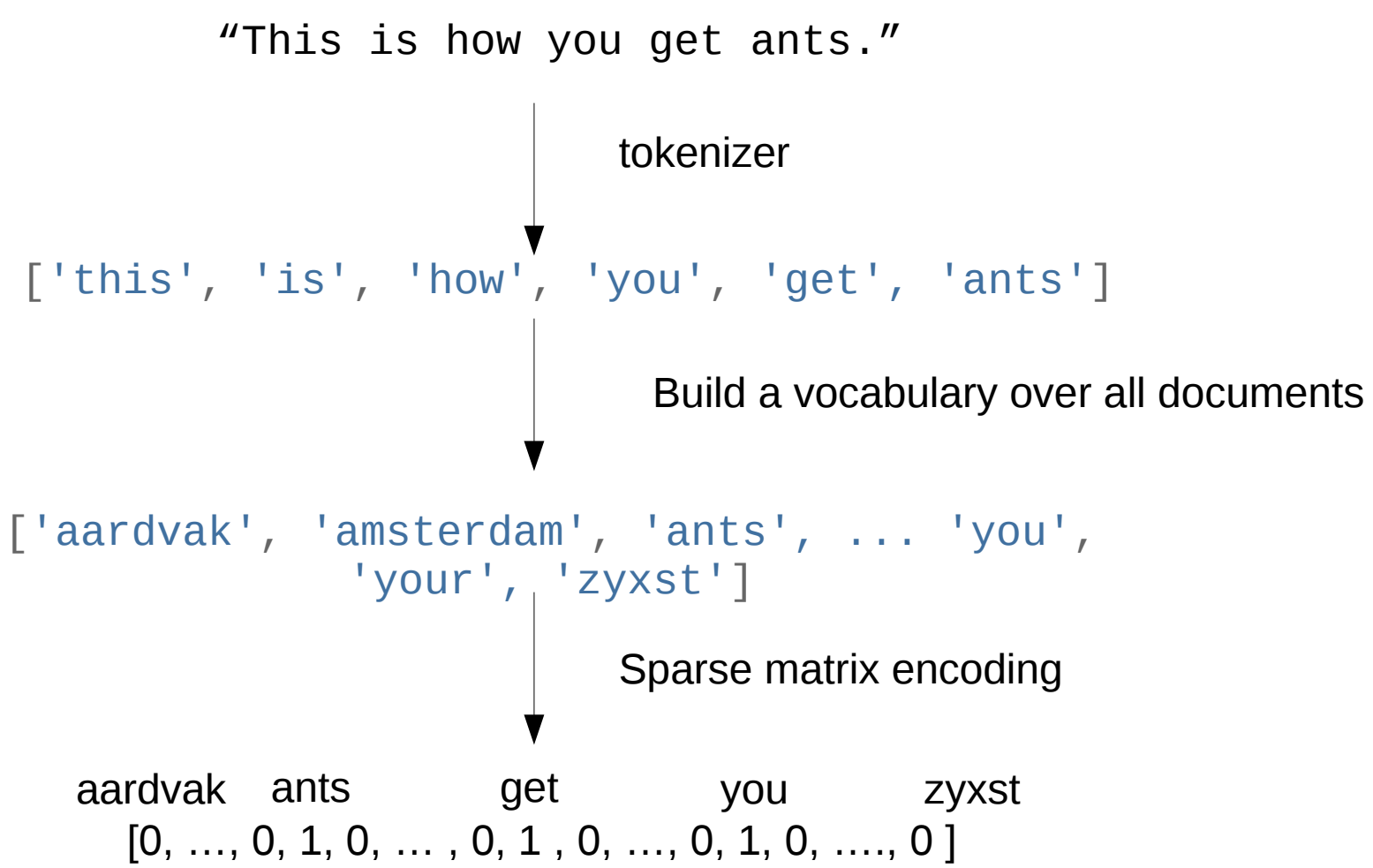
IPython Notebook:
Part 11 – Out Of Core Learning

Stateless Transformers

- `Normalizer`
- `HashingVectorizer`
- `RBFSampler` (and other kernel approx)

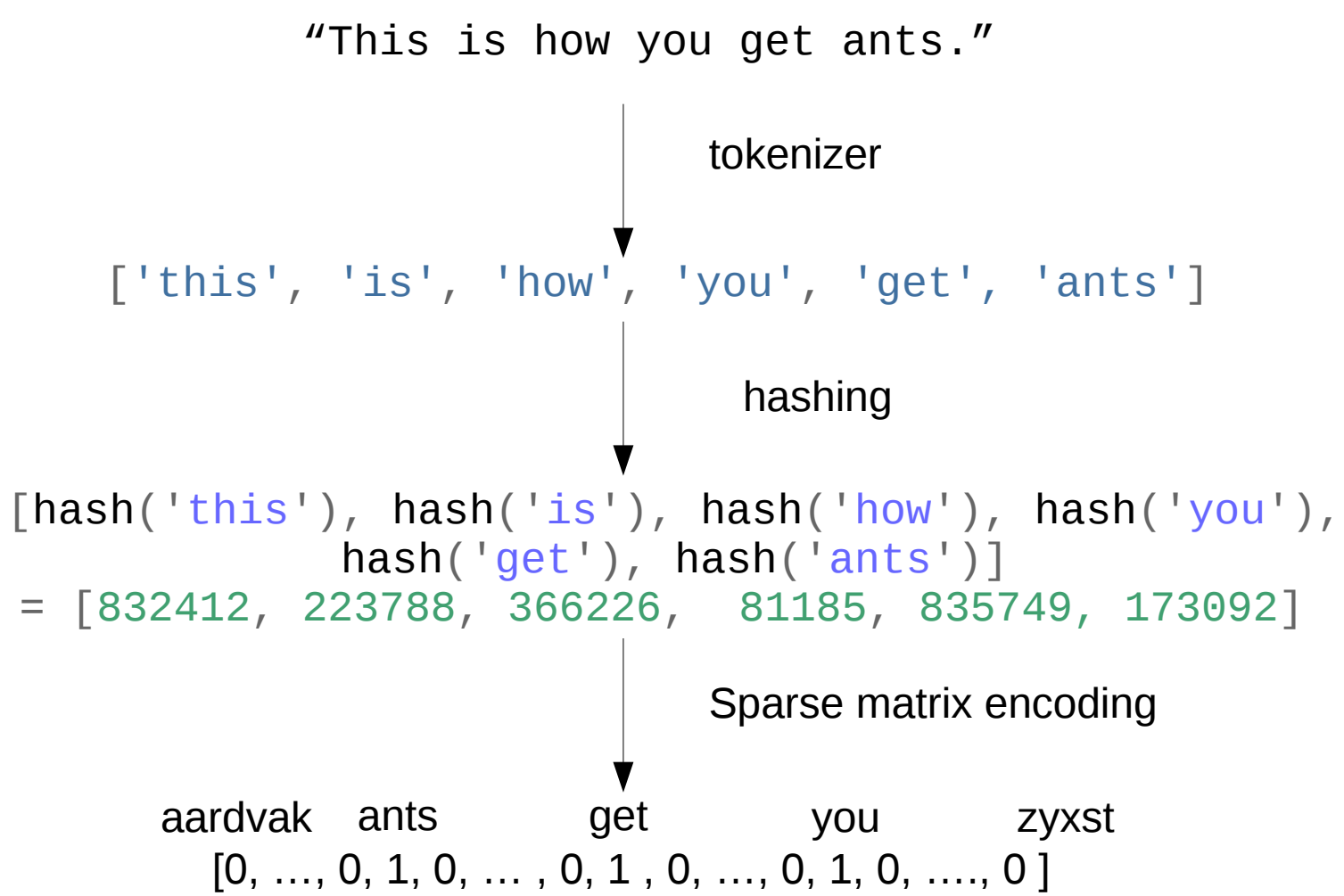
Bag Of Word Representations

`CountVectorizer / TfidfVectorizer`



Hashing Trick

`HashingVectorizer`



IPython Notebook:
Part 12 – Out Of Core Learning for Text

Home Installation Documentation Examples

Google™ Custom Search Search ×

Documentation of scikit-learn 0.16-git

Quick Start

A very short introduction into machine learning problems and how to solve them using scikit-learn. Introduced basic concepts and conventions.

User Guide

The main documentation. This contains an in-depth description of all algorithms and how to apply them.

Other Versions

- scikit-learn 0.15 (stable)
- scikit-learn 0.16 (development)
- scikit-learn 0.14
- scikit-learn 0.13
- scikit-learn 0.12
- Older versions ▲

Tutorials

Useful tutorials for developing a feel for some of scikit-learn's applications in the machine learning field.

API

The exact API of all functions and classes, as given by the docstrings. The API documents expected types and allowed features for all functions, and all parameters available for the algorithms.

Additional Resources

Talks given, slide-sets and other information relevant to scikit-learn.

Contributing

Information on how to contribute. This also contains useful information for advanced users, for example how to build their own estimators.

Flow Chart

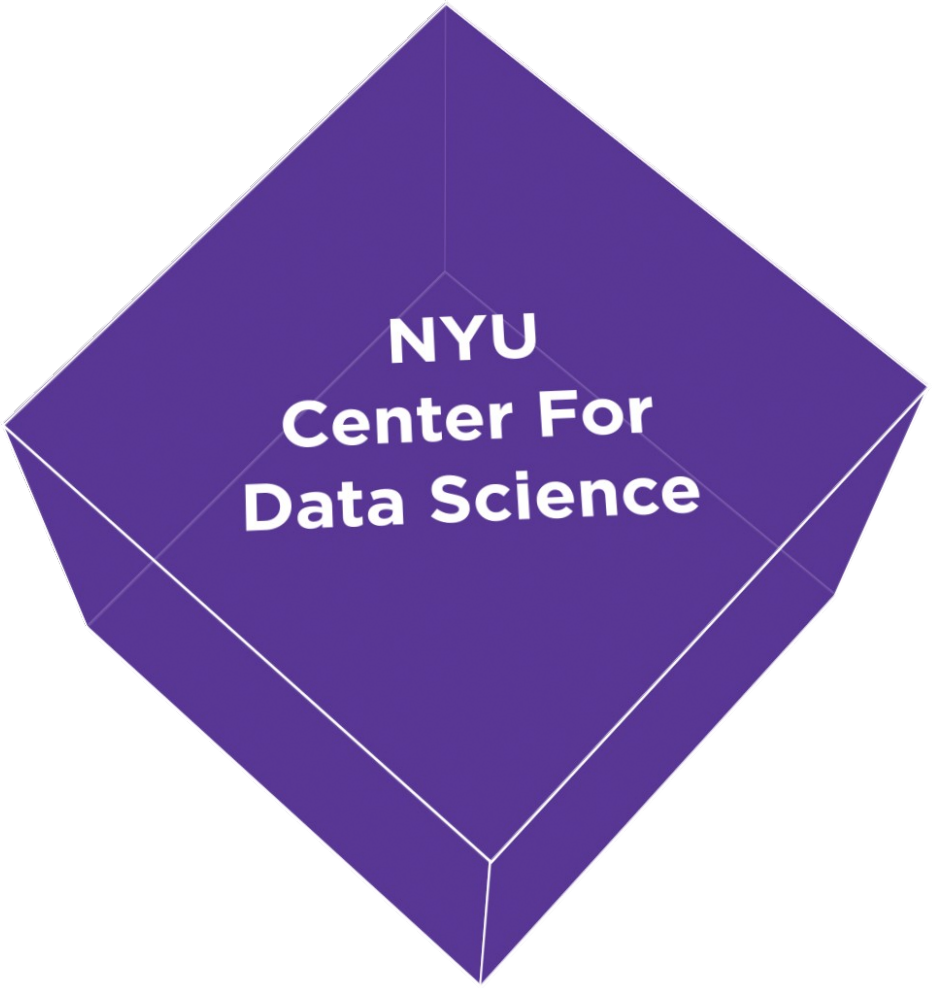
A graphical overview of basic areas of machine learning, and guidance which kind of algorithms to use in a given situation.

FAQ

Frequently asked questions about the project and contributing.

http://scikit-learn.org/

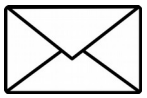
CDS is hiring Research Engineers



Thank you for your attention.


@t3kcit


@amueller


importamueller@gmail.com


http://amueller.github.io